1992年9月1日発行（毎月1回1日発行）第11巻9号通巻125号　昭和58年11月2日第三種郵便物認可

PERSONAL COMPUTER MAGAZINE for MZ, X1, and X68000

Oh!X

特集　数値演算の熱い逆襲

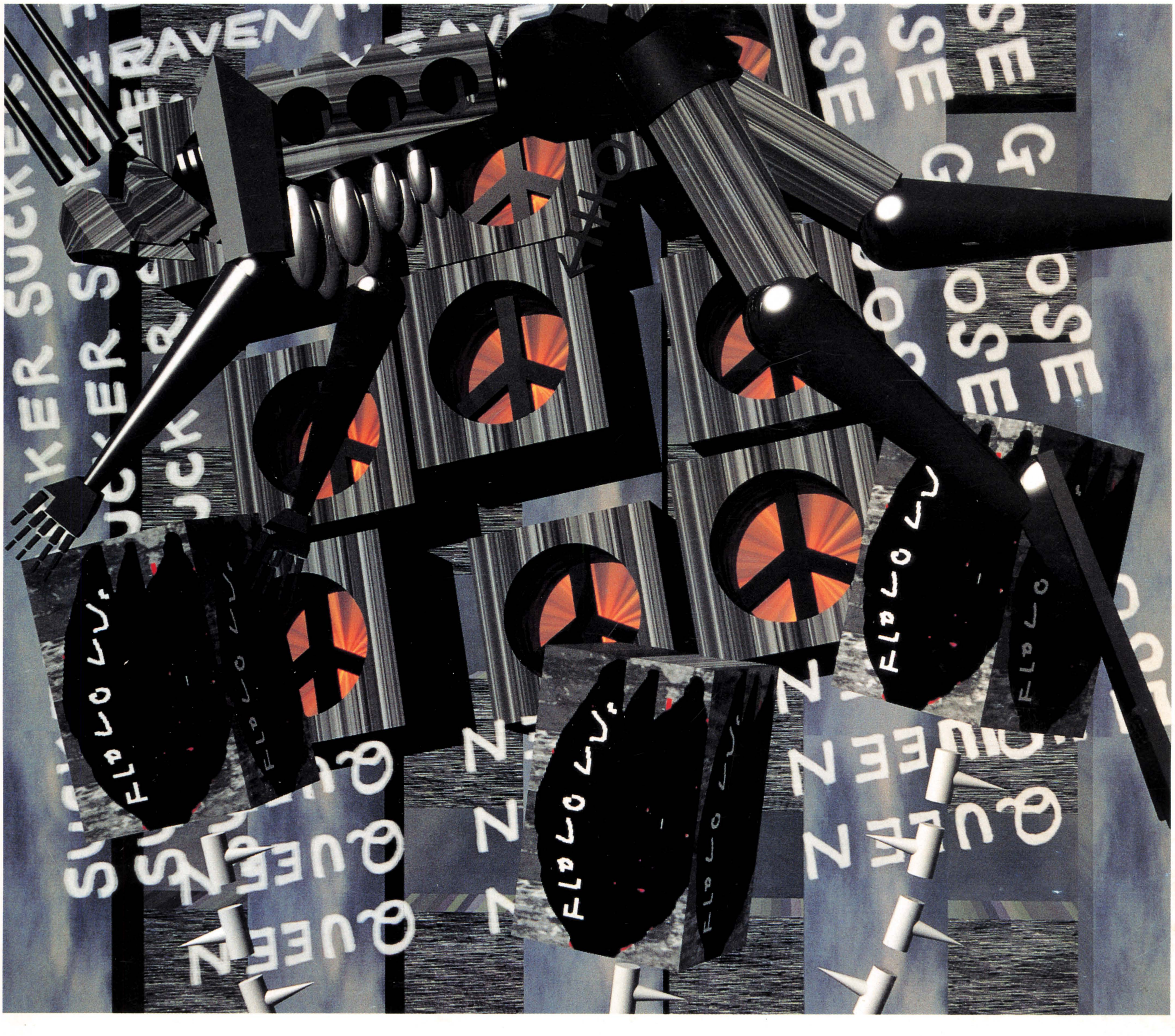
68881の並列駆動/V70による3Dグラフィック/疑似メタボールで遊ぶ

新製品紹介 MATIER/MIRAGE System Model Stuff

Z's-EXジャギーの除去/新連載 DōGA CGアニメーション講座 ver.2.50

9
1992

SOFT BANK　オー／エックス
定価600円



SHARP

目の付けどころが、
シャープでしょ。

“感性”咲かせるワー

POWER WORKSTATION

インテリジェントなパフォーマンスを誇るX68000Compact XVIと
多彩にラインアップされたペリフェラル。感性を刺激するクリエイティブな
ワークステーション環境が自在に構築できます。

- パーソナルワークステーション（2HD3.5インチFDDタイプ・本体＋キーボード＋マウス）
  CZ-674C-H（グレー） 標準価格**298,000**円（税別）
- 15型カラーディスプレイテレビ
  CZ-614D-TN（チタンブラック）・**-BK**（ブラック） 標準価格**135,000**円（税別）
  ■ディスプレイテレビ／CZ-6TU用RGBケーブル**CZ-6CR1** 標準価格**4,500**円（税別）
  ■ディスプレイテレビ／CZ-6TU用TVコントロールケーブル**CZ-6CT1** 標準価格**5,500**円（税別）
- 80MB内蔵用ハードディスクドライブ
  CZ-68HA 9月発売予定
- 5.25インチ増設用フロッピーディスクドライブ
  CZ-6FD5 標準価格**99,800**円（税別・接続ケーブル同梱）
- 光磁気ディスクユニット
  CZ-6MO1 標準価格**450,000**円（税別）
  ■SCSI変換ケーブル**CZ-6CS1** 標準価格**12,000**円（税別）
- 2MB増設RAMボード
  CZ-6BE2D 標準価格**54,800**円（税別・取り付け費別）
  ■2MB増設RAM**CZ-6BE2B** 標準価格**54,800**円（税別・取り付け費別）×2
  ■数値演算プロセッサ**CZ-6BP2** 標準価格**45,800**円（税別・取り付け費別）
- 48ドット熱転写カラー漢字プリンタ
  CZ-8PC5-BK（ブラック） 標準価格**96,800**円（税別）
- MIDIボード
  CZ-6BM1A 標準価格**26,800**円（税別）
- インテリジェントコントローラ
  CZ-8NJ2 標準価格**23,800**円（税別）

X68000 見・体・験フェア ［コンパクト祭り］

この夏をより熱くさせるビッグイベント、X68000見体験フェア「コンパクト祭り」が名古屋に到来。X68000ユーザーはもちろん、パソコンおまかせのキミ、そしてパソコン未体験の人もどしどしご参加ください。――――――――Let's all join in.

●「第1回全日本X68000芸術祭」作品紹介など。

資料請求券
X68000ペリフェラル
Oh!X
9係

クステーション環境。

X68000
PERSONAL WORKSTATION・XVI
Compact

GRAPHIC WORKSTATION

- ●パーソナルワークステーション（2HD3.5インチFDDタイプ・本体＋キーボード＋マウス）
  CZ-674C-H（グレー） 標準価格**298,000**円（税別）
- ●21型カラーディスプレイ **CU-21HD** 標準価格**148,000**円（税別）
- ●80MB内蔵用ハードディスクドライブ **CZ-68HA** 9月発売予定
- ●光磁気ディスクユニット **CZ-6MO1** 標準価格**450,000**円（税別）
  ■SCSI変換ケーブル**CZ-6CS1** 標準価格**12,000**円（税別）
- ●2MB増設RAMボード **CZ-6BE2D** 標準価格**54,800**円（税別・取り付け費別）
  ■2MB増設RAM**CZ-6BE2B** 標準価格**54,800**円（税別・取り付け費別）×2
  ■数値演算プロセッサ**CZ-6BP2** 標準価格**45,800**円（税別・取り付け費別）
- ●カラーイメージスキャナ
  CZ-8NS1 標準価格**188,000**円（税別）
  ■スキャナ用パラレルボード**CZ-6BN1** 標準価格**29,800**円（税別）
- ●カラーイメージジェット
  IO-735X-B（ブラック） 標準価格**248,000**円（税別）
  ■接続ケーブル**IO-73CX** 標準価格**5,500**円（税別）

STANDARD WORKSTATION

- ●パーソナルワークステーション
  （2HD3.5インチFDDタイプ・本体＋キーボード＋マウス）**CZ-674C-H**（グレー） 標準価格**298,000**円（税別）
- ●14型カラーディスプレイ**CZ-608D-H**（グレー） 標準価格**94,800**円（税別）
- ●5.25インチ増設用フロッピーディスクドライブ**CZ-6FD5** 標準価格**99,800**円（税別・接続ケーブル同梱）

TFT COLOR LCD WORKSTATION

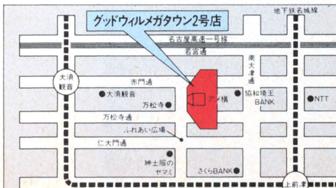
- ●パーソナルワークステーション
  （2HD3.5インチFDDタイプ・本体＋キーボード＋マウス）**CZ-674C-H**（グレー） 標準価格**298,000**円（税別）
- ●10.4型カラー液晶ディスプレイ**LC-10C1-H**（グレー） 標準価格**598,000**円（税別）
  ■接続ケーブル**AN-1515X** 標準価格**4,200**円（税別）

※カラー液晶ディスプレイを接続してご使用の場合、SX-WINDOW上のアプリケーション利用に限定されます。

開催日時：8月21日㈮・22日㈯・23日㈰ 13:00～16:00

会　　場：㈱グッドウィル大須　メガタウン内（名城線上前津駅徒歩3分。）

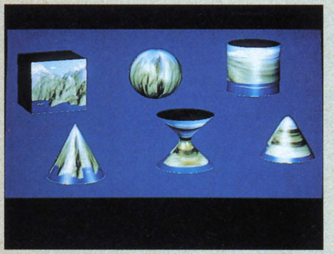
名古屋市中区大須　☎052-242-8581㈹



■主催/グッドウィル■お問い合わせ/シャープエレクトロニクス販売㈱ 中部統轄情報第2営業 ☎052-323-5145㈹ 担当・森

■お問い合わせは…電子機器事業本部システム機器営業部 〒545 大阪市阿倍野区長池町22番22号 ☎(06) 621-1221（大代表）
電子機器事業本部AVCシステム事業推進室 〒162 東京都新宿区市谷八幡町8番地 ☎(03)3260-1161（大代表） シャープ株式会社

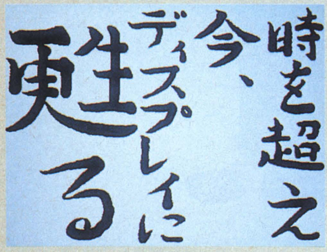
MATIER



MIRAGE System Model Stuff

ファイナルファイト

ライジングサン



(影)のショートプロぱーてぃ

THE USER'S WORKS

Oh!X

CONT

●特集

73 **数値演算の熱い逆襲**

74 微積分をシミュレートする
夏休みの最小2乗法 御木徳高

76 モデル化による演算高速化
疑似メタボールで遊ぶ 丹 明彦

82 V70ボードの活用
AFPPを使った3Dグラフィック 中森 章

90 68881の性能を引き出す
FPP.MACの作成 瀧 康史

99 68881+68881=137762?
68881の並列駆動に挑戦 桒野雅彦

●カラー紹介

36 新製品紹介
MATIERを使う(後編) 川原由唯

65 THE USER'S WORKS
BLUE WINGS

68 新製品紹介
MIRAGE System Model Stuff(2) 丹 明彦

72 特集カラー紹介
数値演算高速化の世界

●THE SOFTOUCH

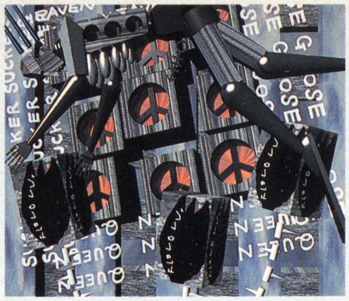
20 SOFTWARE INFORMATION
新作ソフトウェア/TOP10

22 TREND ANALYSIS

GAME REVIEW
24 ファイナルファイト 八重垣那智
28 ライジングサン 影山裕昭
30 ヨーロッパ戦線 金子俊一
32 シューティング68K GAMES 伊譃見あきら

34 AFTER REVIEW
レミングス

〈スタッフ〉
●編集長/前田 徹 ●副編集長/植木章夫 ●編集/岡崎栄子 浅井研二 山田純二 ●協力/有田隆也 中森 章 林 一樹 吉田幸一 華門真人 吉田賢司 影山裕昭 大和 哲 村田敏幸 丹 明彦 三沢和彦 長沢淳博 宮島 靖 金子俊一 浦川博之 石上達也 柴田 淳 御木徳高 ●カメラ/杉山和美 ●イラスト/永沢しげる 山田晴久 寺尾響子 ●アートディレクター/島村勝頼 ●レイアウト/元木昌子 ADGREEN ●校正/グループごじら



表紙絵：塚田　哲也

1992 SEP. 9

E N T S

●シリーズ全機種共通システム

109 THE SENTINEL

110 O-EDIT&MODCNV　黒木淳一

●読みもの

136 X-OVER NIGHT　第26話
新幹線とコンピュータ　高原秀己

138 第63回　知能機械概論―お茶目な計算機たち―
近未来型ワープロ序説　有田隆也

160 猫とコンピュータ　第73回
じいコード，ばあコード　高沢恭子

●連載/紹介/講座/プログラム

40 DōGA CGアニメーション講座 ver.2.50（第1回）
打倒TORNADOへの第一歩（前編）　かまたゆたか

46 大人のためのX68000［第23回］
Communication SX-68K，そしてFIXER　荻窪　圭

50 (影)のショートプロぱーてぃ　その36
書は心を表すものなり　影山裕昭

54 吾輩はX68000である［第16回］
時を刻む　泉　大介

57 X68000マシン語プログラミング Chapter_23H
整数演算のアルゴリズム　村田敏幸

66 響子 in CG わ～るど［第16回］
仮想生物　寺尾響子

117 Z'sSTAFF&Z's-EX用外部ファイル
ジャギー除去に挑戦　御木徳高＆佐藤正春

123 マシン語カクテル in Z80's Bar　第35回
お城と流れ星（その2）　金子俊一

141 Oh!X LIVE in '92
恋をしようよ Yeah! Yeah!（X68000・Z-MUSIC用）　南　正樹
ゆめいっぱい（X68000・Z-MUSIC用）　田所広行
対談!! GMコンポーザー「S.S.T.BAND」　西川善司

148 ハードウェア工作入門〈27〉コンピュータアーキテクチャ編
デジタル論理回路を学ぶ　三沢和彦

153 Creative Computer Music入門（12）
偶成和音と借用和音　瀧　康史

158 ANOTHER CG WORLD　寺尾響子

ペンギン情報コーナー……162
FILES Oh!X……164
Oh!X質問箱……166
STUDIO X……168
編集室から/DRIVE ON/ごめんなさいのコーナー/SHIFT BREAK/microOdyssey……172

UNIXはAT & T BELL LABORATORIESのOS名です。
Machはカーネギーメロン大学のOS名です。
CP/M, P-CPM, CP/Mplus, CP/M-86 CP/M-68K, CP/M-8000, DR-DOSはデジタルリサーチ
OS/2はIBM
MS-DOS, MS-OS/2, XENIX, MACRO80, MS C, MS-WindowsはMICROSOFT
MSX-DOSはアスキー
OS-9, OS-9/68000, OS-9000, MW CはMICROWARE
UCSD p-systemはカリフォルニア大学理事会
TURBO PASCAL, TURBO C, SIDEKICKはBOLAND INTERNATIONAL
LSI CはLSI JAPAN
HuBASICはハドソンソフト
の商標です。その他，プログラム名,CPUは一般に各メーカーの登録商標です。本文中では"TM","R"マークは明記していません。
本誌に掲載されたプログラムの著作権はプログラム作成者に保留されています。著作権上，PDSと明記されたもの以外，個人で使用するほかの無断複製は禁じられています。

■広告目次

アイビット電子……180
アクセス……184
計測技研……181
サンワード……12
J & P……表3
シティソフト……16
シャープ……表2・表4・1・4-9
九十九電機……13
野還ゲームデザイナーズアカデミー……182
パソコンプラザオクト……178・179
ビクター音楽産業……11
P & A……14・15
ブラザー工業……10
マイコンショップ川口……177
満開製作所……176
ラインシステム……183(上)

SHARP

目の付けどころが、
シャープでしょ。

X68000 CompactXVI NEWS

Opinion 1

(ハードディスクが使いたい。)

Compact専用の内蔵ハードディスクが登場します。SCSI仕様の80MB。場所を取らずに高速・大容量ファイル環境を実現します。

■内蔵用ハードディスクドライブ(CZ-674C専用)

CZ-68HA……………………9月発売予定

※取りつけに関してはシャープお客様ご相談窓口にてご相談ください(取りつけ費別)。

さらに大容量をお望みの場合、外付け用のSCSI端子で一般のSCSIハードディスクも接続可能。フルピッチSCSI端子とハーフピッチSCSI端子を接続するためのSCSI変換ケーブルも用意しています。

■SCSI変換ケーブル

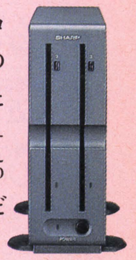
CZ-6CS1………標準価格12,000円(税別)

CZ-6CS1

Opinion 2

(従来のソフト資産を活かしたい。)

これについても、Compact専用の外付け5インチフロッピーディスクユニットを用意していますから、従来の68シリーズの資産を有効活用できます。3.5インチと5インチの間でのデータのやりとりも可能。また、CZ-674C及びCZ-6FD5のスイッチ設定を変えれば、5インチソフトからの起動が可能になり、市販ソフトなどそのまま使えます。



■増設用5インチ・フロッピーディスク・ユニット(CZ-674C専用)

CZ-6FD5……………………標準価格99,800円(税別)

Opinion 3

(ディスプレイテレビを接続したい。)

Compactは、従来のシリーズと比べ体積比44%と小さいため、コネクタの形状も異なっていますが、このケーブルを使用することにより、ディスプレイテレビやRGBシステムチューナーを利用できます。

CZ-6CR1

CZ-6CT1

■15型カラーディスプレイテレビ(スピーカー・チルトスタンド同梱)

CZ-614D-TN…………標準価格135,000円(税別)

■ディスプレイテレビ/CZ-6TU用RGBケーブル

CZ-6CR1……………………標準価格 4,500円(税別)

■ディスプレイテレビ/CZ-6TU用テレビコントロールケーブル

CZ-6CT1……………………標準価格 5,500円(税別)

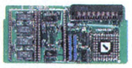
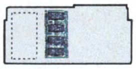
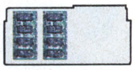
資料請求券
X68000ペリフェラル
Oh!X
9月

パーソナルワークステーション
X68000 Compact XVIについての
ご意見、ご要望にお応えします。

Opinion 4

（メモリ環境をパワーアップしたい。）

Compactは2MBのメインメモリを標準装備していますが、本体内で最大8MBまで拡張できます。

	容量	周辺機器
標準	2MB	——
拡張	4MB	 CZ-6BE2D
	6MB	 CZ-6BE2B
	8MB	 CZ-6BE2B×2

■2MB増設RAMボード　CZ-6BE2D　標準価格54,800円(税別)

■2MB増設RAM　CZ-6BE2B　標準価格54,800円(税別)

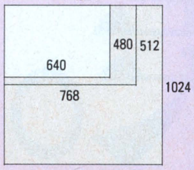
※取りつけに関してはシャープお客様ご相談窓口にてご相談ください(取りつけ費別)。

Opinion 5

（液晶ディスプレイとSX-WINDOWの関係は？）

液晶ディスプレイ(LC-10C1-H 標準価格598,000円・税別)の解像度は640×480ドット。Compactでは、従来のX68000シリーズの画面モードにこの画面モードをプラス。解像度の制約を受けないウィンドウ環境ならではの機能です。このようにSX-WINDOW環境の確立により、ハードウェアに依存しない快適な操作環境が実現します。

SX-WINDOWの実画面エリア
1024×1024ドット
SX-WINDOWの通常表示エリア
768× 512ドット
SX-WINDOW上での
液晶ディスプレイの表示エリア
640× 480ドット



Opinion 6

（数値演算プロセッサはほんとに速い？）

ご存じのようにMPU68000自体は複雑な計算（浮動小数点演算）を単純な計算の組み合わせで行っています。X63000シリーズに装備されている浮動小数点演算パッケージ「FLOAT2.X」は、よく使う単純な組み合わせをまとめたもの。数値演算プロセッサは、いわばこのパッケージの機能を、ハードウェアで高速に実現し、MPUの負担を軽くするものです。アプリケーションプログラムの中には浮動小数点演算を必要としないものもあるため、すべてのプログラムが高速になるわけではありませんが、レイトレーシングなど大量の実数演算を必要とするソフトウェアの場合、飛躍的な実行速度の向上が期待できます。

■数値演算プロセッサ CZ-6BP2 標準価格45,800円(税別)

※数値演算プロセッサはCZ-6BE2D上に装着します。
※取りつけに関してはシャープお客様ご相談窓口にてご相談ください(取りつけ費別)。

X68000
PERSONAL WORKSTATION・XVI
Compact

本体＋キーボード＋マウス
2HD3.5インチFDDタイプ　CZ-674C-H(グレー)　標準価格298,000円(税別)
14型カラーディスプレイ(ドットピッチ0.28㎜)
CZ-608D-H(グレー)　標準価格94,800円(税別)

■お問い合わせは…電子機器事業本部システム機器営業部　〒545　大阪市阿倍野区長池町22番22号　☎(06) 621-1221(大代表)
電子機器事業本部AVCシステム事業推進室　〒162　東京都新宿区市谷八幡町8番地　☎(03)3260-1161(大代表)
シャープ株式会社

SHARP

カラープリンタもスキャナも………

黒の統一美。

画像処理のベストマッチングシステム for X68000。

シャープ株式会社 〒545 大阪市阿倍野区長池町22番22号 ☎(06)621-1221(大代表)

入力から、編集、出力、複写まで、カラーの世界は、シャープです。

SHARP IS COLOR

▶INPUT

X68000用パラレルインタフェースを標準装備した高速コンパクト型イメージスキャナ。

カラーイメージスキャナ JX-220X……標準価格168,000円(税別)

●A4サイズの原稿を約50秒※1で高速読み取り●CCDセンサー採用。さらに中間調処理でシャープでリアルな画像を再現●ディザパターン指定機能※2や濃度補正機能※2など高度な画像処理機能で緻密な読み取りが可能●解像度200ドット/インチ(約7.9ドット/㎜)。ズーム機能で1%きざみの拡大、縮小も可能●色ずれの少ない線順次(1走査)読み取り●X68000シリーズ用「スキャナツール」ソフトを標準装備●プリンタと直接接続することによりダイレクトプリント※3が可能●RS-232Cインタフェース/X68000シリーズ用専用パラレルインタフェースを標準装備。

※1:A4、2値出力、コンピュータへの実転送時間。
※2:表記機能はJX-220X本体使用であり、付属ユーティリティ使用時は異なります。
※3:別売のパラレルインタフェースケーブル(JX-22PC標準価格12,000円(税別)が必要です。

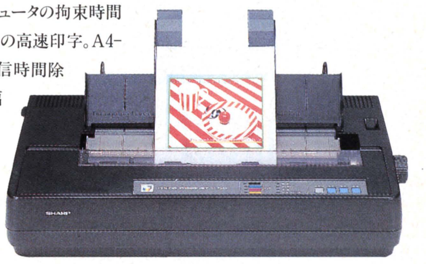
▶OUTPUT

3種類の制御コマンドモードを搭載。質感も鮮やかに再現する高品位カラーイメージジェット。

カラーイメージジェット IO-735X-B……標準価格248,000円(税別)

●シャープ独自のIOシリーズコマンド(Gモード)に加え、NM-9900モード(Nモード)、ESC/P24-84C準拠モード(Pモード)をサポート。一般文書の作成から、各種デザイン、建築用パースなどのCAD分野に対応●発色性に優れた普通紙対応の新黒インキ採用。専用紙はもちろんオフィスでよく使われる普通紙にも鮮明カラー印字●プリントバッファメモリ(128KB)の内蔵で、ホストコンピュータの拘束時間を軽減●48ノズル(各色12ノズル)採用の高速印字。A4-1ページを※約90秒でプリント(データ受信時間除く)●ビジネス用途に適したB4横用紙幅対応●OHPフィルム(専用)にも鮮明プリント●ノンインパクト方式ならではの静粛印字●インキ補充は簡単、経済的なカートリッジ方式

※261×174㎜領域



IO-735X-B 対応アプリケーション

●SX-WINDOW対応ペイントツール
Easypaint SX-68K
CZ-263GW 標準価格12,800円(税別)

●WYSIWYGを実現、ドローグラフィックソフト
CANVAS PRO-68K
CZ-249GS 標準価格29,800円(税別)

●オリジナリティを活かせるポップアップツール
NEW Printshop PRO-68K ver2.0
CZ-221HS 標準価格20,000円(税別)

●マルチワープロ
Multiword PRO-68K
CZ-225BS 標準価格32,000円(税別)

●高速カード型リレーショナルデータベース
CARD PRO-68K ver2.0
CZ-253BS 標準価格29,800円(税別)

●パソコン通信もできるメモリ常駐型ソフト
Teleportion PRO-68K
CZ-258BS 標準価格22,800円(税別)

●これからの高速通信をサポート
Communication PRO-68K ver2.0
CZ-257CS 標準価格19,800円(税別)

資料のご請求・お問い合わせはコンシューマーセンター／東日本相談室 〒162 東京都新宿区市谷八幡町8番地 ☎(03)3260-1161(大代表)・西日本相談室 〒545 大阪市阿倍野区長池町22番22号 ☎(06)621-1221(大代表)・名古屋相談室 〒454 名古屋市中川区山王3丁目5番5号 ☎(052)332-2611(大代表)・福岡相談室 〒816 福岡市博多区井相田2丁目12番1号 ☎(092)575-2381

SHARP

目の付けどころが、
シャープでしょ。

待望のSX-WINDOW
開発支援ツール、登場。

NEW

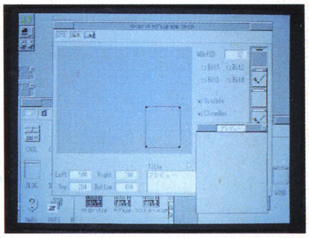
SX-68K

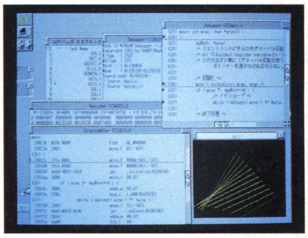
SX-WINDOW 開発キット

CZ-288LWD 9月発売予定

SX-WINDOW用のソフト開発に必要な開発ツールやサンプルプログラムを装備。プログラムの編集、リソースの作成、コンパイル、デバッグといった一連の作業をSX-WINDOW上で効率よく実行できます。初めてSX-WINDOW用のプログラムに挑戦する人にも、簡単に基本機能の理解ができる33種のサンプルプログラム付き。また各マネージャ解説と関数リファレンスの詳細なマニュアルも装備しています。

※本ソフトのご使用に際しては、メインメモリ4MB以上、SX-WINDOW ver2.0以上、C compiler PRO-68K ver2.0以上が必要です。





キット構成

■開発ツール

●SXデバッガ

SX-WINDOW上で複数のプログラムを同時にデバッグすることができるソースコードデバッガ。

●リソースエディタ

SX-WINDOW上のリソースをリソースタイプごとの編集ウィンドウでビジュアルに作成・編集が可能。

●リソースリンカ

Cコンパイラやアセンブラで作成したリソースデータファイル(オブジェクトファイル)をリンクしてリソースファイルを作成。

●サンプルメイク

サンプルプログラムのコンパイル作業をSX-WINDOW上から、XC ver2のMAKE. Xを呼び出して、自動実行する簡易メイクユーティリティ。

■サンプルプログラム

●基礎編(23種)

各マネージャの基本的な機能のみを用いた基本動作の理解。

●応用編(4種)

基礎編での基本機能を応用した簡単なアプリケーションの作成。

●実用編(6種)

基礎/応用編での機能を駆使した、実用的なアプリケーションの作成。

■その他のファイル

●インクルードファイル

Cコンパイラとアセンブラ用の関数定義、データ定義ファイル。

●ライブラリファイル

Cコンパイラ用の関数ライブラリ。

マニュアル

●ユーザーズマニュアル●プログラマーズマニュアル●ファンクションリファレンス●ライブラリリファレンス

X68000
PERSONAL WORKSTATION・XVI
Compact

本体+キーボード+マウス
2HD3.5インチFDDタイプ
CZ-674C-H(グレー) 標準価格298,000円(税別)

14型カラーディスプレイ(ドットピッチ0.28mm)
CZ-608D-H(グレー) 標準価格94,800円(税別)

●5.25インチ増設用
フロッピーディスクドライブ
CZ-6FD5
標準価格99,800円・税別
〔接続ケーブル同梱〕

●ディスプレイテレビ/CZ-6TU用RGBケーブル
CZ-6CR1 標準価格4,500円・税別

●ディスプレイテレビ/CZ-6TU用テレビコントロールケーブル
CZ-6CT1 標準価格5,500円・税別

●SCSI変換ケーブル CZ-6CS1 標準価格12,000円・税別

資料請求券
CZソフト
Oh!X
9係

開いてくださいウィンドウ、触れてくださいインテリジェンス。
さらに拡がる、ＳＸワールド。

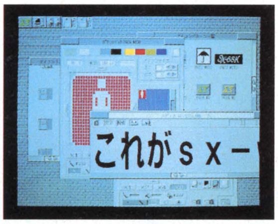
●アウトラインフォント対応、ひらかれたウィンドウ環境。

SX-WINDOW ver2.0

CZ-287SS 標準価格12,800円(税別)

フォントマネージャを装備して待望のアウトラインフォントに対応。画面スクロール機能により、表示画面よりワイドなデスクトップ空間を駆使。アプリケーションのハンドリングに便利なシンボルトレイやアイコンメンテ、パターンエディタなど便利機能満載。

※SX-WINDOW ver1.0(CZ-259SS)およびSX-WINDOW ver1.1(CZ-278SS)をお持ちの方には有償バージョンアップを行います。



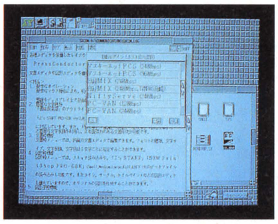
●マルチタスク機能をはじめ、通信環境がさらに充実。

Communication SX-68K

NEW

CZ-272CWD 8月発売予定

通信環境をさらに高めたウィンドウ対応の通信ソフトです。マルチタスク機能により他のアプリケーションソフトを実行中でも簡単に通信が可能。また、ホスト局をクリックするだけの自動ログイン機能、初心者にも簡単なプログラム機能、最新モデム(20種類)もフルサポートしています。



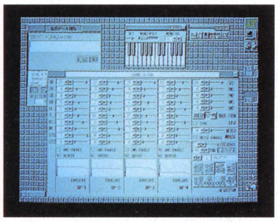
●多彩なサウンドクリエイトを実現するFM音源サウンドエディタ。

SOUND SX-68K

NEW

CZ-275MWD 8月発売予定

他のミュージックソフトで演奏中の音色を、簡単に作成・変更ができるマルチタスク機能、またエディット、イメージ、ウェーブの3つの編集/確認モードを装備。作成中の音色も50曲の自動演奏でリアルタイムに確認、編集できます。まさにミキサー感覚で音創りが楽しめるツールです。

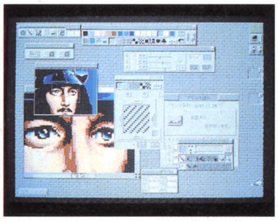


●ウィンドウ対応グラフィックツール。

Easypaint SX-68K

CZ-263GWD 標準価格12,800円(税別)

マウスによる簡単操作、65,536色中16色の多彩な表現、クリエイティブマインドに応えるウィンドウ対応ペイントツールです。同時に複数のウィンドウを開いて編集でき、各ウィンドウ間でのデータ交換もできます。



※SX-WINDOW対応ソフトの動作には、メインメモリ2MBおよびSX-WINDOW ver1.1以上が必要です。

充実のPROシリーズ

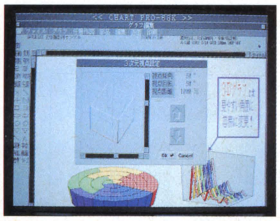
●ビジネスグラフチャート

CHART PRO-68K

NEW

CZ-267BSD 標準価格38,000円(税別)

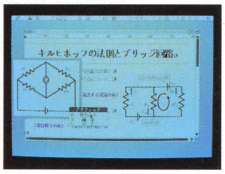
各種データベースで作成したデータをもとに、多彩なグラフが作成できます。3次元表示やグラフの複合機能も装備。データはMultiword,Press Conductor PRO-68Kに取り込むこともできます。



●グラフィック機能搭載の本格派ワープロ

Multiword ver1.1

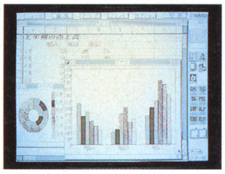
CZ-225BSD 標準価格32,000円(税別)



●簡単操作の統合型表計算ソフト

BUSINESS PRO-68K Popular

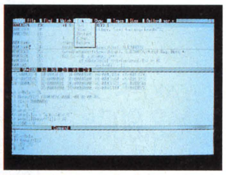
CZ-286BSD 標準価格28,000円(税別)



●各種ドライバ、ライブラリを追加

COMPILER PRO-68K ver2.1

CZ-285LSD 標準価格44,800円(税別)

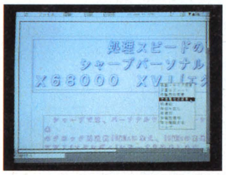


※有償バージョンアップ対応中。

●各種エディタ装備のレイアウトソフト

PressConductor PRO-68K

CZ-266BSD 標準価格28,000円(税別)



※以上のPROシリーズのソフトの動作にはメインメモリ2MB必要です。

※発売予定のソフトの画面写真は実物とは異なる場合があります。

●お問い合わせは…シャープ㈱電子機器事業本部AVCシステム事業推進室 〒162 東京都新宿区市谷八幡町8番地 ☎(03)3260-1161(大代表)へ。 シャープ株式会社

brother

パソコンソフト自販機 TAKERU

コンテスト入賞作品いよいよ登場!!

SHOOTING 68K GAMES
シューティング68K

アモルファスが作りあげた強力シューティングゲームツール"シューティング68K"。このツールでユーザーが創りあげたゲームのコンテストが行なわれたのは知ってるよネ!応募総数350点の作品の中から厳重な審査を経て選ばれたグランプリ1作品と優秀賞2作品がTAKERUから出るぞっ!アイデア・センス・技術ともに秀でた作品たちを、ぜひその目で確かめてほしい。

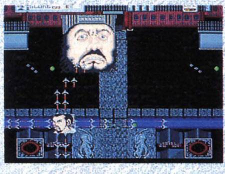
8月20日発売! TAKERU価格¥各3,000(税込)

※下記の3作品が1本ずつ発売されます。
※「シューティング68Kゲームコンテスト」はマイコンベーシックマガジン誌上で行なわれたものです。

―グランプリ―

「ヴァリストレスナルト」

★優秀賞2作品★



「三国志―幻伝―」&「フェイバー・ザ・ロード」

チェイスH.Q.
CHASE H.Q.

好評発売中

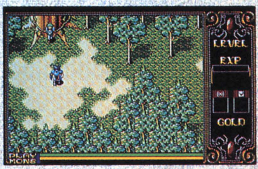
TAKERU価格 ¥7,800(税込)

©TAITO

TAKERU価格 ¥6,800(税込)

©1989/1990 マイクロキャビン

8月下旬発売予定



ブラザー工業株式会社 〒467 名古屋市瑞穂区苗代町2番1号
TAKERU事務局 (052)824-2493
東京営業所 (03)5203-7133
大阪営業所 (06)252-4234

通信販売 通信販売をご希望の方は、ソフト名・機種名・住所・氏名・電話番号を明記の上TAKERU事務局まで現金書留でお申し込み下さい
代金引換は一度、現金書留で申し込んで頂いた方に案内させて頂きます

Victor

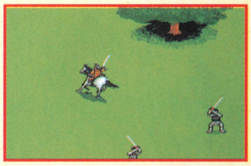
野望が渦巻く!!

貴族の世界か武士の時代か‥‥源氏と平家の、壮絶な戦い。

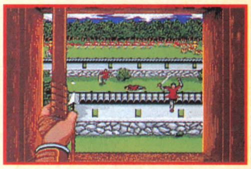
平安時代末期、新たに台頭してきた武士「源氏」と、朝廷までも自己の権力の道具にした貴族「平家」との壮絶な権力闘争を、リアルタイム・シミュレーションで贈る話題作ついに登場。

プレイヤーは「源 頼朝」か「源 義経」になって源氏の軍勢を指揮して平家軍と戦います。平家を全滅させるか、全国に築かれた城を制圧するかして日本を統治します。リアルタイムで行われる「合戦」や「馬追い」「攻城戦」「忍者戦」などの変化にとんだ４つのアクション・ゲームが、従来のシミュレーション・ゲームにないドラマ性を加味した全く新しいタイプのゲームになっています。米国生まれの、そして米国で大評判を呼んだシネマウエア社の「ローズ・オブ・ザ・ライジング・サン」をバージョンアップして発売。

日本全体マップで
各地の動向を把握。


馬追いのアクション
敵の武将を倒せ!

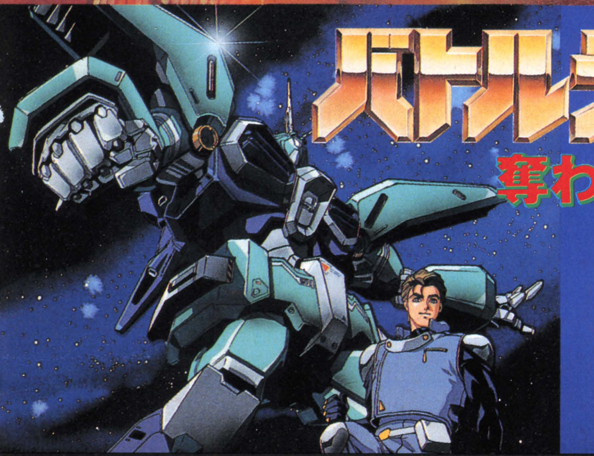
忍者戦は、手裏剣との戦いだ。


敵が城を攻撃 一兵たりとも城中に
入れないよう弓で倒せ。

行軍の開始だ。

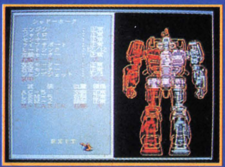
合戦は兵をうまく
配置して戦術を練れ。

© 1992 VICTOR MUSICAL INDUSTRIES, INC.



奪われた聖杯を取り戻せ!
広大な宇宙空間に展開されるロボット・ウォーズ!

■アメリカで爆発的なロボットブームを巻き起こした話題作いよいよ登場!
■3Dポリゴンの採用による迫真のバトル・アクション。
■プレイヤーが自らコックピットに乗り込みロボットを操縦、リアルなロボット・シミュレーションを体験。
■共に戦うクルーとして41人の傭兵から最大3人までの傭兵の採用により、戦略性もゲームの重要な要素。
■情報収集と賞金稼ぎによってグレイドの高いメックを手にいれて、150の惑星を舞台に任務を遂行。



好評
発売中!!
X68000対応
(5FD, 3.5FD)
¥9,800(税別)

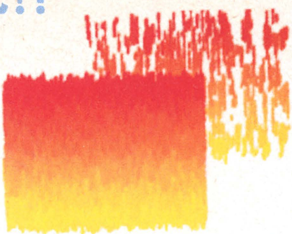
© 1992 Activision. All rights reserved. Battletech is registered trademarks of FASA Corporation. © 1992 Victor Musical Industries, Inc.

発売元：ビクター音楽産業株式会社

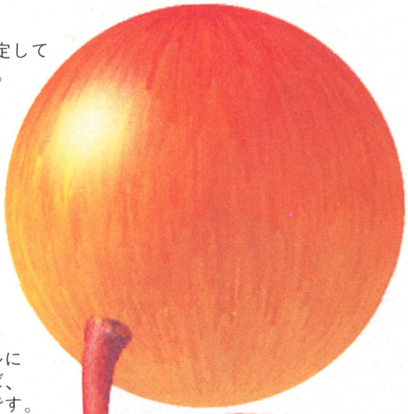
通信販売 当社の商品をお近くのパソコンショップでお買い求めになれない場合、商品名、機種名、住所、氏名、電話番号を明記のうえ、下記住所まで定価プラス3%消費税分を現金書留にてお申し込み下さい。(送料無料) 〒151 東京都渋谷区千駄ヶ谷2-8-16 ビクター音楽産業㈱〈通信販売係〉

何がすごいか！

2次元なのに3次元!?



1 リンゴの模様を自動ペインティング機能で一気に作成します。



2 光源やハイライトを設定して球状に３Ｄ変形します。

3 最後に、メッシュコントロールによる自由変形機能で変形すれば、簡単にリアルなリンゴの完成です。

X68000がグラフィックワークステーションに変身!
マチエールは、プロのデザイナー集団がつくったプロのためのペイントソフトです。高い機能と使いやすいマウスオペレーションを実現しました。

大画面編集も思いのままに

512×512ドット標準画面の解像度では、フィルム出力をして印刷物にするには不足です。マチエールではメモリー増設により最大2048×2048ドットの画面をリアルタイムに編集できます。

複数の画面で快適編集

512×512ドットを同時に最大4画面までもつことができます。(メモリ2Mバイト時は2画面まで)絵のパーツを作っておいたり、2つの画面を合成したり、クリエイティブワークの能率が大幅アップします。画面間の便利な合成機能もいろいろ用意してあります。

立体文字の作成も簡単

どんな図形も簡単に立体表現することができます。「書体倶楽部」(Zeit社)のアウトラインフォントや、スキャナでとりこんだロゴマークなども、マチエールで立体文字にすれば、ビジュアル効果も抜群です。

ディザでフルカラーを実現

マッハバンドのない美しいグラデーションは、角度・増減率とも自由に設定可能、4隅の色指定もできます。
ぼかし・3次元表示など高度な画像処理も1670万色フルカラーで実現しました。

ジャギーのない高品質

拡大・縮少・変形・パース変形・メッシュ変形など、すべてオーサンプリングによるジャギーのない高品質を実現しました。

多彩な編集機能

コピー・クリップコピー・シェードコピー・タイルコピー・拡大・縮小・変形・回転・パース変形・メッシュ自由変形・円筒マッピング・球面マッピング・領域交換・矩形スクロール・ミラー反転・各種マスク機能

専用ソフトなみの画像処理機能

ネガ反転・ディフューズ・ぼかし・モノクロ化・二値化・ランダムノイズ・平滑化・鮮鋭化・輪郭抽出・レリーフ・モザイク・フレア・コントラスト補正・色変換など

使いやすいスキャナー入力

スキャナ原稿台のプレビュー表示をマウスで範囲指定する簡単操作。

高機能なプリンタ出力

画面の任意の範囲を、最大A3までの自由なサイズでプリントアウトできます。

■ 対応画像フォーマット・入出力機器

画像ファイル形式	PIC・GL3・GLX・IMG・RGB・TIFF(Mac・TOWNS互換)
カラープリンター	SHARP IO-735X・SHARP CZ-8PC3・SHARP CZ-8PC5・NEC PC-PR406・HP Desk Jet 505
モノクロプリンタ	Canon BJ-10V・ESC/P系・PC-PR系
ビデオプリンタ	SHARP CZ-6PV1・NEC PC-VC101
ビデオ取り込み	SHARP CZ-6VT1
カラースキャナー	SHARP CZ-8NS1・SHARP JX-220X(以上純正パラレルボード対応)・EPSON GT-4000・EPSON GT-6000
タブレット	WACOM SD-510C

■ おまけソフト　アニメーションツール「うごくZO」・スクリーンセーバー・画像ユーティリティ

■ 監修　CGデザイナー　長谷川 一光

新登場

9月上旬発売予定

プロ仕様ペイントツール

Hyper Image Processor

Matier

商標登録申請済

マチエール

対応機種 X68000(要2M)　価格39,800円(税別)

株式会社 サンワード
SUNWARD
〒213 川崎市高津区下作延1043 TEL(044)855-4335

親切と安さのTSUKUMO

30TH YEAR SINCE 1962

ツクモの日まつり

9/1〜9/20まで

掲載商品2万円以上送料無料(一部地域を除く)

ツクモグローバルカード

大人気！入会者募集中！

18才以上なら学生さんもOK！

国内・外で活躍！使って便利、持ってて安心！ツクモグローバルカードはジャックス・VISAとの提携カードです。ツクモ各店でのお買物がらくらくできる上に、国内はもとより海外での分割ショッピングもOK！
20才以上の方にはキャッシングカードも発行致します。
お申し込みは☎03(3251)9898又は店頭にて！

7/17札幌に新店 OPEN！

DEPOツクモ2番街店

是非ご来店ください

☎011(242)3199

ツクモの日 周辺機器まつり!!

VIP-120CX
定価￥112,000
29%off
特価￥79,000
19台限り

アンフィニーシステム
MIC-68K
定価￥24,800
特価 ￥19,900

Canon バブルジェットプリンタ
BJ-10VC+専用プリンターケーブル
特価￥49,000

CZ-8NS1 定価￥188,000
9台限り
42%off
特価￥109,000

シャープよりいろんなパソコン！どれを選ぶ？

ビジネスでバリバリ持ち歩く方へ
DOS/V対応のOADG仕様SLノート
PC-6700シリーズがお勧め！
PC-6781J
定価￥630,000
3.5" 1.44MB FDD1基・
80MB HDD内蔵

ツクモ特価販売中！

ワープロユースが中心で更にDOS/Vマシンのソフトを使う方へ
「書院パソコン」がお勧め
PC-WD1シリーズ
定価￥330,000より

ツクモ特価販売中！

X68000 SUPER-HDセット
●CZ-623C-TN
定価￥448,000
＋
●14型カラーディスプレイ
9セット限り

セット特価￥269,000

ツクモの日目玉品

PC-9801NS/Tセット
●PC-9801 NS/T(本体)
定価￥288,000
●120MBハードディスク
定価￥128,000
合計定価￥416,000
ツクモの日特価￥298,000

超低金利冬・夏ボーナス二回払い受付中！お問い合せください。

ツクモはSONY MOディスク正規販売代理店です

これが今一番の人気者！
SONY 3.5インチ光磁気ディスクユニットセット
●RMO-S350(3.5" 光磁気ディスクドライブ)………￥235,000
●SCSIケーブル………￥6,900
●SCSIインターフェースボード………￥29,800
合計定価 ￥271,700
ツクモ特価販売中！
シャープ純正「CZ-6MO1」も特価販売中

一流メーカーHDD

X68000用
SCSIハードディスク
100MB
ツクモ特価 ￥69,000
200MB
ツクモ特価￥110,000
※SCSIボード(CZ-6BSI 定価￥29,800)は別売です。

TSドライブ(X68000用)

目のつけどころがツクモでしょ。
<仕様>●3.5インチ2DD/2HD/2HCフォーマット対応の為、いろいろなフォーマットのメディアを読み書きができます。●ユーティリティソフト付属(ディバイスドライバー／フォーマッター)
※初代X68KはROM交換が必要です。

X68000シリーズ専用3.5インチフロッピーディスクドライブ
TS-3XRシリーズ
3.5インチ1ドライブ **TS-3XR1** 定価￥44,800
特価￥35,800(消費税別途￥1,074)
3.5インチ2ドライブ **TS-3XR2** 定価￥57,800
特価￥46,800(消費税別途￥1,404)
※只今開発中。X68000CompactXVI用外付け5インチFDD

コンピュータミュージック(X68000用)

NEW Aセット
●CM-32L………￥69.000
●SX-68M-II………￥19.800
●Musicstudio Mu-1 Ver1.4………￥19.800
合計定価￥108,600
ツクモ特価￥88,000
(消費税別途￥2.640)
クレジット例(18回払・税込)
初回￥5,863＋月々￥5,600×17回

NEW Bセット
●CM-300………￥58.000
●SX-68M-II………￥19.800
●Mu-1 SUPER………￥39.800
合計定価￥117.600
ツクモ特価￥92,000
(消費税別途￥2.760)
クレジット例(10回払・税込)
初回￥10,919＋月々￥10,000×9回

NEW Cセット
●CM-500………￥115.000
●SX-68M-II………￥19.800
●Mu-1 SUPER………￥39.800
合計定価￥174.600
ツクモ特価￥141,000
(消費税別途￥4.230)
クレジット例(15回払・税込)
初回￥11,300＋月々￥10,500×14回

NEW Dセット
●CM-64………￥129.000
●SX-68M-II………￥19.800
●Mu-1 SUPER………￥39.800
合計定価￥188.600
ツクモ特価￥154,000
(消費税別途￥4.620)
クレジット例(18回払・税込)
初回￥10,261＋月々￥9,800×17回

※この他の組み合わせは、お問い合わせ下さい。☎03-3251-9911へ
※Mu-1 Ver1.4は3.5インチのメディアはありませんのでご注意下さい。

ローランド 追加オプション機器
ステレオマイクロモニター CS-10………定価￥~~17,000~~
MIDIキーボードコントローラー PC-200………定価￥~~36,000~~

コンピュータアート

▶INPUT◀
JX-220X
A4サイズカラーイメージスキャナー………定価￥~~168,000~~

Z' STAFF PRO-68K Ver.3.0 &対応タブレットセット
WACOM製
SD-510C タブレット………￥98,000
TJ-410A-2 接続ケーブル………￥6,000
SP-200 スタイラスペン………￥10,000
ツァイト Z" STAFF PRO-68K Ver.3.0………￥58,000
合計定価 ￥172,000
ツクモ特価 ￥145,000

▶OUTPUT◀
■IO-735X-B
カラーイメージジェットプリンター………定価￥~~248,000~~
■BJC-820J
カラーバブルジェットプリンター………定価￥~~398,000~~
■CZ-8PC5-BK
カラー熱転写プリンター………定価 ￥~~96,800~~

メモリーボード

■1MB増設RAMボード(CZ-600C専用)
ツクモ特価￥19,500
■1MB増設RAMボード(ACE/PRO/PRO2シリーズ用)
ツクモ特価￥17,000
■2MB増設RAMボード(拡張スロット用)
ツクモ特価￥33,800
■4MB増設RAMボード(拡張スロット用)
ツクモ特価￥59,800
※計測技研のメモリーボードも取り扱っております。価格についてはお尋ね下さい。

パソコン通信

時代は9600ボーへ!!
■モデム 9600bps/MNP5 & CCITT V. 42bis
ツクモ特価￥69,800
■通信ソフト た〜みのる2
ツクモ特価￥14,000

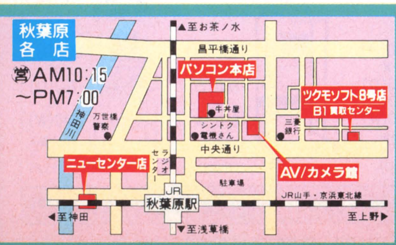
電子文具

シャープ 電子マネジメント手帳
タイムマネージメントを管理する便利ツール新登場！従来の電子システム手帳用ICカードがそのまま使えます。次から次へと、忙しい方への為の強力な助っ人。大画面・大容量・手書き入力で操作効率向上！※更に、便利な名刺読み取り機「PV-BR1」もお勧めです。
PV-F1 標準価格￥128,000 ツクモ特価販売中！

全国どこからでも通話料無料

通信販売のご注文は下記フリーダイヤルへ。
受・注・専・用フリーダイヤル **0120-377-999**
通販センター **03-3251-9911** 商品についてのお問い合わせは各店又は通販へ。

クレジット払い	各種リース払い	現金書留払い
月々￥3,000以上の均等払いも頭金なし、夏・冬ボーナス2回払いも受付中！	くわしくは各店にお問い合わせ下さい。ケースに合わせてご相談にのります！	〒101-91 東京都千代田区神田郵便局私書箱135号 ツクモ通販センター Oh！X係
カード払い	**全国代金引き換え配達**	**銀行振込払い**
通信販売での御利用カード、ツクモグローバルカード、VIPカード、セントラル、ジャックス※御本人様より電話で通販部へお申し込み下さい。	お申し込みは☎03-3251-9911へお電話1本！配達日の指定もできます。	事前に☎でお届け先をご連絡下さい。三和銀行 秋葉原支店(普)1009939 ツクモデンキ



ツクモは「スーパーX PRO SHOP」です。

PRO STAFF ツクモ

パソコン本店 荒井

九十九電機㈱
〒101-91 東京都千代田区神田郵便局私書箱135号
★商品のご注文は在庫確認の上お願いします。★表示価格には消費税は含まれておりません。

ツクモパソコン本店2F
☎03-3253-1899(直通)
パソコン本店代表 ☎03-3253-5599 (担当／荒井) 休毎週木曜

■ツクモニューセンター店 ☎03-3251-0987(担当／沢栄)休毎週木曜
■名古屋1号店 ☎052-263-1655(担当／吉高)休毎週火曜(但し9/15は営業)
■名古屋2号店 ☎052-251-3399(担当／横山)休毎週水曜
■ツクモ札幌店 ☎011-241-2299(担当／田口)休毎週木曜
■DEPOツクモ2番街店 ☎011-242-3199(担当／四条)休毎週木曜

※定休日が祝日と重なる場合は営業致します。

P&Aならではの新品パソコン 5年保証

《業界No.1の"P&Aメンテナンスサポート》
最高の保証システム
①業界最長の新品パソコン5年保証
（※モニター・プリンター３年間保証!! ※一部商品は除きます。）
②中古パソコンの１年間保証
（モニター・プリンター６ヶ月間保証）
③初期不良交換期間3ケ月
（※新品商品に限らせていただきます）
④永久買取保証
⑤配達の指定OK!!
⑥夜間配送もOK!!
（※PM6:00〜PM8:00の間 ※一部地域は除きます。）

便利でお得な支払いシステム
①翌月一括払い手数料無料（ご利用下さい。）
②業界No.1の低金利
③月々の支払いは￥1,000より
④9ヶ月先からのスキップ払いOK!!
⑤84回までの分割、ボーナス併用OK!!
⑥カレッジクレジット
⑦ステップアップクレジット
⑧ボーナスだけで10回払いOK!!
⑨現金一括払いOK!!
（※商品・金額ご確認の上、銀行振込・現金書留にてご入金下さい。）

またまたまた

8/18〜9/17

《増設メモリー&数値演算プロセッサ》計測技研

1 PRKII-02(2M)……定価￥ 55,000▶特価￥ 39,800
2 PRKII-04(4M)……定価￥ 90,000▶特価￥ 67,000
3 PRKII-06(6M)……定価￥125,000▶特価￥ 92,500
4 PRKII-08(8M)……定価￥160,000▶特価￥119,000
5 PRKII-12(2M)……定価￥ 85,000▶特価￥ 63,000
⑥PRKII-14(4M)……定価￥120,000▶特価￥ 89,500
⑦PRKII-16(6M)……定価￥155,000▶特価￥114,500
⑧PRKII-18(8M)……定価￥190,000▶特価￥141,000
⑨MC-68881RC……定価￥ 38,000▶特価￥ 27,000

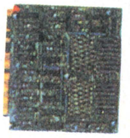
カラーイメージジェット
■IO-735X-B
定価￥248,000
特価￥152,000
（送料・消費税込み￥157,590）

■Z's STAFF PRO 68K Ver.3.0
（ツアイト）（定価￥58,000）
特価 ￥36,500
（送料・消費税込み￥38,110）

■SX-68MII（MIDI）
（サコム）定価￥19,800
特価￥13,500
（送料・消費税込み￥14,420）

■HGS-68（スキャナ）
（HAL研）定価￥39,800
特価￥24,500
（送料・消費税￥26,265）

X68000メモリボード（I/O・DATA）（送料￥500）



①SH-6BE1-1M（600CE用）定価￥25,000
（送料・消費税込み￥18,952）·特価￥17,900
②PIO-6BE1-A 定価￥25,000
（送料・消費税込み￥16,583）··特価￥15,600
③PIO-6BE2-2M 定価￥50,000
（送料・消費税込み￥32,239）··特価￥30,800
④PIO-6BE4-4M 定価￥88,000
（送料・消費税込み￥55,620）··特価￥53,500

FDD（5インチ×2基）
■CZ-6FD5
（シャープ）（定価￥99,800）
P&A超特価!!
TEL下さい。

注目!!冬のボーナス一括払い手数料(金利)無料(9月末/10月末/11月末/12月末のいずれかをご指定下さい。)

X68000 CompactXVI/XVI/XVI-HD

※送料￥2,000、消費税別

今月の特選!! 特価品

■Compact XVI さらにお安くなります。

●CZ-674C-H
●CZ-608D-H
●CZ-6FD5（5"FDD）
定価￥492,600
P&A超特価~~￥320,000~~
（※X68000サービスゲーム全て付いています。）
（モニターをCZ-606Dに変更の場合￥10,000を引いて下さい）

右記セットでお買い上げの方にもれなくプレゼント!!
①「ダウンタウン熱血物語（￥8,800）」
はもちろん、さらにその上、人気の
㋑「ロードス島戦記（￥9,800）」
㋺「グラディウスII（￥9,800）」
㋩「ザ・プロサッカー68（￥9,800）」
㋥「信長の野望武将風雲録（￥9,800）」
㋭「ELLE（エル）（￥7,800）」
の中のいずれか2本をプレゼント!!

X68000-CompactXVI ●ディスケット10枚●ジョイカード2ケプレゼント中!! さらにお安くなります。
Ⓐセット：CZ-674C＋CZ-608D……定価￥392,800▶特価~~￥281,000~~

12回	24,700	24回	13,000	36回	9,000	48回	7,000	60回	5,900

X68000-XVI▶セットでお買い上げの方に●ディスケット10枚●ジョイカード2ケプレゼント中!!
Ⓐセット：CZ-634C-TN＋CZ-606D-TN……定価￥447,800▶特価価格はTEL下さい。

12回	26,200	24回	13,800	36回	9,600	48回	7,500	60回	6,300

Ⓑセット：CZ-634C-TN＋CZ-614D-TN……定価￥503,000▶特価価格はTEL下さい。

12回	29,700	24回	15,700	36回	10,800	48回	8,200	60回	7,100

X68000-XVI-HD▶セットでお買い上げの方に●ディスケット10枚●ジョイカード2ケプレゼント中!!
Ⓐセット：CZ-644C-TN＋CZ-606D-TN……定価￥597,800▶特価価格はTEL下さい。

12回	35,700	24回	18,800	36回	13,000	48回	10,200	60回	8,600

Ⓑセット：CZ-644C-TN＋CZ-614D-TN……定価￥653,000▶特価価格はTEL下さい。

12回	39,000	24回	20,600	36回	14,300	48回	11,200	60回	9,400

※上記のモニターを、CZ-606D（定価￥79,800）、CZ-604D（定価￥94,800）、CZ-607D（定価￥99,800）、CZ-605D（定価￥115,000）、CZ-608D（定価￥94,800）、CZ-614D（定価￥135,000）、CU-21HD（定価￥148,000）に変更の場合、TEL下さい。超特価で販売致します。

X68000シリーズ〜P&Aスペシャルセット

（送料￥2,000・消費税別）

注目 スペシャルプレゼント!!
★ SUPER-HD には、
上記の①をプレゼント
★ PRO-II には、上記の
①＋㋑〜㋭の中の2本をプレゼント
ズバリ価格で大奉仕中
●ディスケット10枚、●ジョイカード2個プレゼント中

SUPER-HD P&A特選セット 限定

Ⓐセット：■CZ-623C-TN（単品）……………定価￥498,000▶特価￥199,000
Ⓑセット：■CZ-623C-TN＋CZ-606D……定価￥577,800▶特価￥254,000
Ⓒセット：■CZ-623C-TN＋CZ-608D……定価￥592,800▶特価￥267,000
Ⓓセット：■CZ-623C-TN＋CZ-607D……定価￥597,800▶特価￥269,000
Ⓔセット：■CZ-623C-TN＋CZ-614D……定価￥633,000▶特価￥289,000
Ⓕセット：■CZ-623C-TN＋CU-21HD……定価￥646,000▶特価￥299,000

PRO-II P&A特選セット 限定

Ⓐセット：■CZ-653C（単品）……定価￥285,000▶特価￥138,000
Ⓑセット：■CZ-653C＋CZ-606D……定価￥364,800▶特価￥195,000
Ⓒセット：■CZ-653C＋CZ-604D……定価￥379,800▶特価￥197,000
Ⓓセット：■CZ-653C＋CZ-608D……定価￥379,800▶特価￥207,000
Ⓔセット：■CZ-653C＋CZ-607D……定価￥384,800▶特価￥209,000
Ⓕセット：■CZ-653C＋CZ-614D……定価￥420,000▶特価￥229,000
Ⓖセット：■CZ-653C＋CU-21HD……定価￥433,000▶特価￥239,000

X68000用ハードディスク

①LHD-FM100E（ロジテック）
定価￥99,800
▶P&A超特価TEL下さい。
②LHD-FM200E（ロジテック）
定価￥138,000
▶P&A超特価TEL下さい。
③HD-J100（システムサコム）（￥128,000）………
（送料・消費税込み￥87,550）…▶特価￥ 84,000
④HD-J130（システムサコム）（￥148,000）
（送料・消費税込み￥104,030）…▶特価￥100,000
⑤HD-J170（システムサコム）（￥189,000）
（送料・消費税込み￥121,540）…▶特価￥117,000

プリンター（ケーブル用紙付）（送料￥1,000 消費税別）

■CZ-8PC5-BK 定価￥ 96,800▶特価￥69,000
■CZ-8PK10…定価￥ 97,800▶特価￥71,000
■CZ-8PG2…定価￥160,000▶特価価格はTEL
■CZ-8PG1…定価￥130,000▶特価価格はTEL

モデム

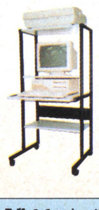
■PV-M24B5
（AIWA）（定価￥39,800）
▶特価￥25,000
（送料・消費税込み￥26,780）
■MD-24FB5V
（オムロン）（定価￥39,800）
▶特価￥25,500
（送料・消費税込み￥27,295）
■FMMD-311G
（富士通）（定価￥35,800）
▶特価￥24,800
（送料・消費税込み￥26,574）

P&A特選パソコンラック（消費税別）（送料無料）

①3段￥7,900

1230(H)×600(D)×650(W)
（消費税込み￥8,137）

②4段￥8,800



1230(H)×600(D)×650(W)
（消費税込み￥9,064）

③5段￥12,500

1310(H)×700(D)×640(W)
（消費税込み￥12,875）

全機種＝移動自由（キャスター付）●5段のみ＝キーボード収納可能、電源コード付（2.5m）（2P）

●本広告の掲載の商品の価格については、消費税は含まれておりません。 ●営業時間＝平日AM10:00〜PM7:00、日祭AM10:00〜PM6:00

アフターサービス万全
全商品保証付。専門の担当がお客様の立場で対応します。
初期不良、輸送トラブルetc.
万が一初期不良、輸送トラブルが発生しました際には、即交換させていただきます。

★頭金なし!!
★即日発送!!

秋葉原でおなじみのP&Aがズバリ超特価セールでご奉仕!!

- ●お近くの方は、お立寄下さい。専門係員が説明いたします。
- ●本体単品でも受付します。詳しくは、お電話にてお問合せ下さい。
- ●ビジネスソフト定価の15%引きOK.!! TEL下さい。
- ●現金書留及び銀行振込でお申し込みの方は、上記商品の料金に3%加算の上でお申し込み下さい。詳しくは、お電話でお問い合せ下さい。

全国通販

X68000用ソフトコーナー （送料1ヶ～5ヶまで￥500・消費税別）

商品	定価	特価
◆Z's STAFF PRO68K Ver3.0(ツアイト)	定価￥58,000	特価￥36,500
◆Z's TRIPHONY デジタルクラフト(ツアイト)	定価￥39,800	特価￥27,000
◆テラッツォ(ハミングバード)	定価￥19,400	特価￥13,600
◆マジックパレット(ミュージカルプラン)	定価￥19,800	特価￥14,200
◆Gツール(ザインソフト)	定価￥28,000	特価￥18,600
◆たーみのる2(SPS)	定価￥17,800	特価￥13,000
◆Mu-1 Super	定価￥39,800	特価￥28,500
◆サイクロンEXPRESSα68	定価￥98,000	特価￥69,000
◆KAMIKAZE(サムシンググッド)	定価￥68,000	特価￥43,800
◆C-TRACE68 Ver3.0(キャスト)	定価￥98,000	特価￥68,500
◆G68K Ver.2 PRO	定価￥22,000	特価￥17,300
◆C&Professional Pack(マイクロウェアジャパン)	定価￥58,000	特価￥39,800
◆Final Ver3.2(エーエスピー)	定価￥38,000	特価￥29,000
◆CZ-213MSD MUSIC PRO68K	定価￥18,800	特価￥13,200
◆CZ-214MSD SOUND PRO68K	定価￥15,800	特価￥11,300
◆CZ-215MSD Sampling PRO68K	定価￥17,800	特価￥12,500
◆CZ-220BSD DATA PRO68K	定価￥58,000	特価￥40,000
◆CZ-224LSD The 福袋 Ver2.0	定価￥ 9,900	特価￥ 7,400
◆CZ-225BSD Multiword Ver1.1	定価￥32,000	特価￥23,000
◆CZ-243BSD CYBERNOTE PRO68K	定価￥19,800	特価￥15,000
◆CZ-247MSD MUSIC PRO68K[MIDI]	定価￥28,800	特価￥20,500
◆CZ-249GSD CANVAS PRO68K	定価￥29,800	特価￥22,000
◆CZ-251BSD Hyper word	定価￥39,800	特価￥29,400
◆CZ-253BSD CARD PRO68K Ver2.0	定価￥29,800	特価￥22,700
◆CZ-257CSD Communication PRO68K Ver2.0	定価￥19,800	特価￥15,300
◆CZ-258BSD Teleportion PRO68K	定価￥22,800	特価￥16,900
◆CZ-261MSD MUSIC studio PRO68K Ver2.0	定価￥28,800	特価￥21,200
◆CZ-263GWD Easypaint SX-68K	定価￥12,800	特価￥ 9,800
◆CZ-265HSD New Print Shop Ver2.0	定価￥20,000	特価￥15,400
◆CZ-266BSD Press Conductor PRO68K	定価￥28,800	特価￥22,000
◆CZ-284SSD OS-9/X68000 Ver2.4	定価￥35,800	特価￥25,600
◆CZ-285LSD C-Compiler PRO68K Ver2.1	定価￥44,800	特価￥32,500
◆CZ-286BSD BUSINESS PRO68K Popular	定価￥28,000	特価￥20,500
◆CZ-287SS SX-WINDOW Ver2.0	定価￥12,800	特価￥ 9,800

☆ゲームソフト25%OFF OK.!!（一部ソフト除く）

周辺機器コーナー （送料￥500・消費税別）

商品	定価	特価
①CZ-8NS1	定価￥188,000	特価￥133,000
②CZ-6VT1	定価￥ 69,800	特価￥ 49,500
③CZ-6TU	定価￥ 33,100	特価￥ 23,900
④BF-68PRO	定価￥ 19,800	特価￥ 14,400
⑤CZ-8NM3	定価￥ 9,800	特価￥ 7,200
⑥CZ-8NT1	定価￥ 13,800	特価￥ 10,000
⑦CZ-6BE2A	定価￥ 59,800	特価￥ 42,800
⑧CZ-6BE2B	定価￥ 54,800	特価￥ 39,300
⑨CZ-6BE2D	定価￥ 54,800	特価￥ 39,300
⑩CZ-6BF1	定価￥ 49,800	特価￥ 35,800
⑪CZ-6BP1	定価￥ 79,800	特価￥ 57,000
⑫CZ-6BM1	定価￥ 26,800	特価￥ 19,300
⑬CZ-6EB1	定価￥ 88,000	特価￥ 63,000
⑭AN-S100	定価￥ 36,600	特価￥ 26,300
⑮CZ-6SD1	定価￥ 44,800	特価￥ 32,500
⑯CZ-6BN1	定価￥ 29,800	特価￥ 21,500
⑰CZ-6BV1	定価￥ 21,000	特価￥ 15,200
⑱CZ-6BC1	定価￥ 79,800	特価￥ 57,000
⑲CZ-6BG1	定価￥ 59,800	特価￥ 43,000
⑳CZ-6BU1	定価￥ 39,800	特価￥ 28,500
㉑CZ-6PV1	定価￥198,000	特価￥142,000
㉒CZ-6BS1	定価￥ 29,800	特価￥ 21,500
㉓CZ-8NJ2	定価￥ 23,800	特価￥ 17,500
㉔CZ-6BL2	定価￥298,000	特価￥214,000
㉕JX-100S	定価￥ 89,800	特価￥ 44,000
㉖JX-220X	定価￥168,000	特価￥121,000
㉗IO-735XB	定価￥248,000	特価￥154,000
㉘LC-10C1H	定価￥598,000	特価￥459,000
㉙CZ-6CS1(674C用)	定価￥ 12,000	特価￥ 8,900

中古・高価現金買取り 下取りOK!!

■まずはお電話下さい。
下取り専用
買取り電話 ▶ ☎03-3651-1884 FAX. 03-3651-0141
■下取り・買取りで、お急ぎの方は、直接当社に来店、または宅急便にてお送り下さい。

買取り価格…完動品・箱/マニュアル/付属品付の価格です。

- ●下取りの場合……価格は常に変動していますので査定額をお電話で確認して下さい。（差額は、P&A超低金利クレジットをご利用下さい。）
- ●買取りの場合……現品が着き次第、2日以内に買取り金額を連絡し、振込み、又は書留でお送り致します。
- ●近郊の方は、P&A本店まで、直接お持ち下さい。即金にて、￥1,000,000までお支払い致します。

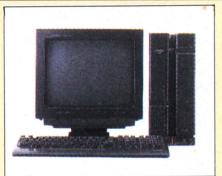
- ●最新の在庫情報・価格はお電話にてお問い合せください。
- ●買い取りのみ、または、中古品どうしの交換も致します。詳しくは電話にて、お問い合せ下さい。
- ●価格は変動する場合もございますので、ご注文の際には必ず在庫をご確認下さい。
- ●本商品の掲載の価格については、消費税は、含まれておりません。
- ●現金書留及び銀行振込でお申し込みの方は、上記商品の料金に3%加算の上でお申し込み下さい。詳しくは、お電話でお問い合せ下さい。

《便利な超低金利クレジットをご利用下さい》

●月々￥1,000円からOK.!! ●ボーナス払いOK（夏冬10回までOK）
●支払い回数 1回～84回 ●お支払いは、8ヶ月先からでもOK.!!

●定休日/毎週水曜日

P&A特選＝今月の中古特選品



●CZ-601C
●CZ-611D-TN
￥120,000

●CZ-634C-TN
●CZ-606D-TN
￥248,000

●CZ-644C-TN
●CZ-604D-TN
￥318,000

買取り価格

機種	価格	機種	価格
●CZ-634C	￥170,000	●CZ-602C	￥75,000
●CZ-644C	￥230,000	●CZ-612C	￥85,000
●CZ-604C	￥100,000	●CZ-652C	￥55,000
●CZ-623C	￥138,000	●CZ-662C	￥75,000
●CZ-603C	￥ 85,000	●CZ-611C	￥68,000
●CZ-613C	￥105,000	●CZ-601C	￥45,000
●CZ-653C	￥ 75,000	●CZ-600C	￥45,000
●CZ-663C	￥ 90,000		

下取り交換差額表

新品 / 下取り	CZ-634C モニターセット	CZ-644C モニターセット	モデル UX20セット	モデル CX20セット	9801DA2
CZ-623C モニターセット	150,000	270,000	70,000	160,000	130,000
CZ-613C モニターセット	190,000	290,000	100,000	190,000	160,000
CZ-652C モニターセット	230,000	340,000	150,000	240,000	220,000
CZ-604C モニターセット	180,000	290,000	100,000	190,000	160,000
CZ-600C モニターセット	230,000	340,000	150,000	240,000	220,000

注目!!冬のボーナス一括払い手数料(金利)無料（9月末/10月末/11月末/12月末のいずれかをご指定下さい。）

通信販売お申し込みのご案内

〔現金一括でお申し込みの方〕
●商品名およびお客様の住所・氏名・電話番号をご記入の上、代金を当社まで、現金書留でお送りください。（プリンター・フロッピーの場合、本体使用機種名を明記のこと）

〔銀行振込でお申し込みの方〕
●銀行振込ご希望の方は必ずお振込みの前にお電話にてお客様のご住所・お名前・商品名等をお知らせください。
（電信扱いでお振込み下さい。）

〔振込先〕 住友銀行 新小岩支店
普通預金 1451576 ㈱ピー・アンド・エー

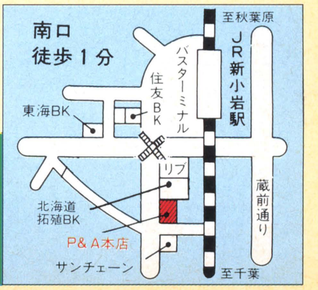
〔クレジットでお申し込みの方〕
●電話にてお申し込みください。クレジット申し込み用紙をお送りいたしますので、ご記入の上、当社までお送りください。
●現金特別価格でクレジットが利用できます。残金のみに金利がかかります。
●1回～84回払いまで出来ます。但し、1回のお支払い額は￥1000円以上。

超低金利クレジット率

回　数	3	6	10	12	15	24	36	48	60	72
手数料	3.0	4.0	5.5	5.5	8.5	11.5	16.0	21.0	27.0	33.0

マイコン専門ショップ
P&A
株式会社ピー・アンド・エー
〒124 東京都葛飾区新小岩2丁目1番地19号
☎03-3651-0148（代） FAX. 03-3651-0141

営業時間
平日：AM10:00～PM7:00
日祭：AM10:00～PM6:00



●価格は流通事情により変動致しますので、銀行振込・書留等の送付前に、あらかじめお電話にてご確認下さい。

City Soft X68000 FEP RECALL

新バージョン

SHARP X68000

日本語入力プロセッサ

FIXER ver.4.0

FPコール対応版

価格：19,800円

FIXER Ver.4.0は構文意味解析処理による高い変換効率を実現。多彩な変換モード（逐次自動変換・一括変換・高速変換・句読点による変換）を装備して使用用途に合った環境を提供。キー操作も標準添付（ASK68K）のFEPと上位互換になっているので、買ったその日から違和感なく使用できます。
さらに辞書の圧縮・拡張・編集などを行えるユーティリティプログラムも添付されています。

バージョンアップ内容

- ASK標準のFPコール準拠対応。
  WP等のアプリケーション上でも使用が可能になります。
- キーカスタマイズ
  変換に使用するキーの配置が自由に設定できます。
- SX-WINDOW上での速度アップ
- CTRL＋アルファベットキーへの機能割り付け（BS等）
- ローマ字変換の追加（WI：ウィ、WE：ウェ）

変換効率の良さで定評の高い“FIXER Ver4.0”X68000版がキーアサイン、FPコール対応により、日本語ワープロ等より多くのアプリケーションで御使用いただけるようになりました。

技術は夢から生まれる

Citysoft

シティソフト株式会社
〒534 大阪市都島区善源寺町2-7-5
TEL.06-927-1060 FAX.06-927-1067

※広告の内容は変更することがあります。

新雑誌「月刊PC」では「92年度ベストパソコン大賞」のユーザー投票を行います！

1992年10月17日。ソフトバンクから"モノを主役としたまったく新しい大型パソコン総合誌"「月刊PC」がデビューします。その創刊を記念して、大々的にユーザー投票を行い、92年度の「ベストパソコン大賞」を選出します。今年、数々のパソコン、ソフト、周辺機器のなかで、あなたは何を選び、何に感動しましたか？　ユーザー１人１人のメッセージの集結が、よりよいパソコンを育てます。新雑誌PCの新しい試みにぜひ、参加してください。

新雑誌「PC」の特徴

「PC」。それはもちろんPersonal Computerの頭文字です。16ビットパソコンが誕生して10年。パーソナルコンピュータの市場は複数のアーキテクチャが勢力を競い合うなかで、量的拡大・多様化に拍車がかかり、いよいよ混沌としてきました。もはやユーザーには、自分が使っていくパソコンの真の姿が見えにくくなってきています。

いまどんなパソコンを選べば良いのか。パソコン関連製品を購入していくユーザーに、必要な製品情報を提供する"パーソナルコンピューター総合情報誌"として、「PC」がデビューします。

「PC」の主な特徴は以下の通りです。●毎号、購入可能なパソコン関連製品の情報をすべて網羅。●話題のパソコン、ソフト、周辺機器を100ページ以上でレビュー。●投稿ページを充実。ホビーストたちの参加型メディアです。●気軽に読めるエッセイ、ゲーム記事などエンタテイメントも充実。●写真がキレイでビジュアルな誌面構成。

ベストパソコン大賞

PCでは、その年に発売されたパソコン関連製品のベストワンを決める、「ベストパソコン大賞」を行っていきます。第１回目の今回は、92年に発売されたすべてのパソコン関連製品のなかから、読者アンケートの集計結果、そして集計データをベースにPCの一流のレギュラー評者たちによる選考会を行い、パソコン、ソフト、プリンタ、ハードディスクなど、各ジャンルの92年度ベストプロダクトを選定していきます。

なお、ユーザー投票による「ベストパソコン大賞」のノミネート発表は10月17日発売の創刊号で、各ジャンルのベストワンを決める部門賞およびベストパソコン大賞発表は1993年１月号（12月18日発売）で行います。

アンケートには豪華特典付き！

裏面のアンケート項目をご覧いただき、ハガキの裏に答を記入し、62円切手を貼ったうえで投函お願いします。

投票していただいた方には豪華特典として、抽選の上ベストパソコン大賞に選ばれたパソコンをはじめ、プリンタ、ハードディスク、ワープロ、表計算など各種周辺機器やアプリケーションソフトのモニターを募集します。発表は93年１月号で行います。20名以上の方にチャンスがありますので、どんどんご応募ください。

投票〆切り日は９月１日です！
（当日の消印有効）

郵便はがき

お手数ですが
62円切手を
お貼り下さい

〒108

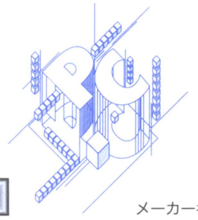
東京都港区高輪２-19-13
NS高輪ビル

ソフトバンク（株）出版事業部
月刊 **PC** 編集部　行

お名前	年齢　才	性別　男性　女性
勤務先または学校名（学部）		
職種および役職（技術、営業、事務、ほか）		
ご住所　〒		
TEL		
ご使用のパソコン 自宅 勤務先・学校		
パソコン経験年数	趣味	
併読雑誌ベスト３（一般誌含む）		
モニターを希望するパソコン関連製品（例ハードディスクなど）		

PCが選ぶ、92年ベストパソコン大賞!

アンケート解答欄



1

	メーカー名	製品名		購入価格
●パソコン本体				
●周辺機器	メーカー名	製品名		購入価格
・プリンタ				
・ハードディスク			容量	
・光磁気ディスク				
・増設メモリ			容量	
・ディスプレイ			インチ	
・モデム			通信速度	
・その他				
●ソフトウェア	・ メーカー名	製品名	メディア	購入価格
・ワープロ				
・表計算				
・データベース				
・統合				
・通信				
・言語				
・グラフィックス				
・エディタ				
・ユーティリティ				
・その他				
	メーカー名	製品名		購入価格
●サプライ製品				

2

	製品名	理由
・パソコン本体		
・周辺機器		
・ソフトウェア		
・サプライ		

	メーカー名	製品名	理由
3			
4			
5			
6			
7			

8

	メーカー名	理由
・ハードウェア		
・ソフトウェア		

9 店頭　カタログ　雑誌広告　パソコン誌（雑誌名）　知人の意見　その他

メッセージ欄

アンケート項目

1 今年購入したパソコン本体、周辺機器、ソフトウェアなどをすべて記入して下さい。
(それぞれメーカー名、製品名、購入価格を明記)

2 購入した製品であなたが最も気に入った製品と、その理由をあげて下さい。

3 今年のパソコン関連製品でいちばん素晴らしいと感じた製品は何ですか（購入の有無にかかわらず）。また、その理由をあげて下さい。

4 今年購入したかったが、なんらかの理由で購入できなかったパソコン関連製品は何ですか。

5 現在発売中のハードウェアで最も優れたデザインだと思う製品名と理由をあげて下さい。

6 現在発売中のソフトウェアで最も優れた画面デザインだと思う製品名と理由をあげて下さい。

7 現在ご使用のパソコン関連製品で最も優れたマニュアルを持つと思う製品名と理由をあげて下さい。

8 現在、最もあなたが期待するメーカーはどこですか。また、その理由をあげて下さい。

9 製品を購入する場合に、参考にする資料は何ですか。

パソコン選びの総合テスト情報誌

PC MAGAZINE パソコン・マガジン

JAPANESE EDITION

特集 1

9月号 毎月18日発売 定価640円(税込)

目的別システム構築術 PC-9800完全武装

ビジネスからホビーまで、PC-9801FA、PC-9801NS/Tをマルチに使いこなす

特集2

PC-9800+DOS/V、Mac これで解決! 異機種間データ共有法

購入ガイド

高速タイプが続々登場!
最新ハードディスク
購入マニュアル

FIRST LOOKS

- 低価格カラーインクジェットプリンタ DeskJet 505J
- Windows用ディスクユーティリティ Norton Desktop
- HDD容量を倍加する圧縮ツール DISKX II

C言語技術情報誌 Cマガジン

9月号 定価980円(税込) 毎月18日発売

C MAGAZINE

提携:COMPUTER LANGUAGE誌　監修:石田晴久

特集 Borland C++ Ver.3.0研究

プロローグ Turbo CからBorland C++へ
Part I BC++の機能
C言語としてのBC++、C++としてのBC++
Part II APPLICATION FRAMEWORKS
Turbo Vision、Object Windows
Part III 新しい統合環境(Turbo C for Windows)
Part IV BC++の今後の展開(BC++3.1)

特別記事

DOS/Vユーザ必読
PC-9801フォーマットをIBM互換機で読む

好評連載

実践Cプログラミング入門
C++入門講座・Try The C++
新MS-DOS プログラミング入門

COMPUTER LANGUAGE誌提携記事

Who's Miuding the Store? Referevce Counting in C++

特別付録 5"2HDディスク

- 1.2M2HD読み込みデバイスドライバ「JAPAN2HD」
- メニュープログラム「GAS」
- テキスト整形プログラム「Efin」
- アウトラインプロセッサ「Ofin」
- 本誌掲載ソースプログラム　ほか

SOFT BANK ソフトバンク出版事業部
〒108 東京都港区高輪2-19-13 NS高輪ビル
TEL:03-5488-1360

月刊PC 10月創刊

モノが主役の、まったく新しいパソコン誌です。

「PC」では、投稿を募集します！

新パソコン情報誌PCは、パソコンユーザー参加型メディアです。
あなたの生の声を、広く全国のパソコンユーザーに伝えます。

READER'S REVIEW

実際にパソコン関連製品を使ってみて、肌で感じた製品の良かった点、ガッカリした点を書いて、他のパソコンユーザーに教えてあげてください。

PC READERS

秋葉原通信、ハードやソフトのオリジナル活用法、サポートの不平不満、長期稼働マシン自慢etc.…。さまざまなコーナーを設けて、あなたの原稿をお待ちしています。

破格の原稿料で、お応えします！

採用分の投稿には原稿料をお支払いいたします。投稿は原稿用紙でもフロッピー（DOSフォーマット）でもOK。ただし原稿は400字詰で7枚までとさせていただきます。

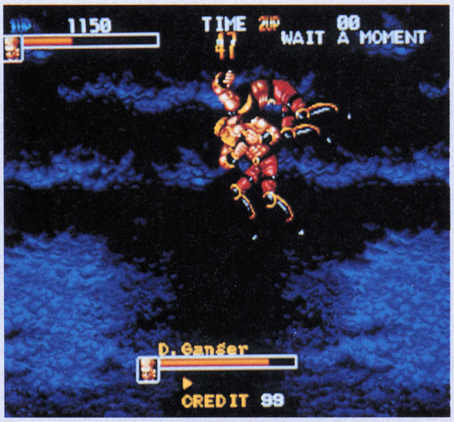
宛先は
〒108 東京都港区高輪2-19-13　NS高輪ビル
ソフトバンク㈱出版事業部　「月刊PC」投稿係

ソフトバンク株式会社
出版事業部　〒108 東京都港区高輪2-19-13 NS高輪ビル
TEL03-5488-1360　FAX03-5488-1364

SOFTWARE INFORMATION

8月28日に「ポピュラスII」が発売される。前作のようにハマる人は多発するのだろうか？　エグザクトの「Etoile Princesse」が9月以降まで発売が延期されてしまったのは、ちょっと残念。

デスブレイド



対戦型格闘ゲーム「デスブレイド」がX68000に移植されることになった。

プレイヤーはまず，8人の闘士のなかから自分の好きなキャラクターを選び出し，次々に現れる敵を倒していく。

レバーを使って奥行きのあるステージを歩き回り，小技ボタンと大技ボタンで攻撃できる。基本的には小技ボタンでパンチ，大技ボタンでキックということになっているが，もちろんボタンとレバーの組み合わせでいろいろな技をかけることができる。エルボースマッシュ，ボディスラム，ロープ振り，アバランシュホールド，アイリッシュホイップ，スープレックス，ネックブリーカー，バックドロップなどなど。

そして，これらの技のほかに，必殺技を出すチャンスも訪れる。体力ゲージの下にOKのサインが出たら，そのタイミング。逃さず攻撃すれば，スゴい技が出せるぞ。

X68000用　　5″2HD版　　価格未定
SPS　　☎0245(45)5777

うまいのはゲーム会社の格闘モノ！

1.	ファイナルファイト	2↑
2.	グラディウスII	1↓
3.	OVERTAKE	9↑
4.	出たな!!ツインビー	10↑
5.	三國志III	5
6.	ジェノサイド2	ー↑
7.	ふしぎの海のナディア	ー↑
8.	スターウォーズ	4↓
9.	シムアース	7↓
10.	大戦略III'90	ー↑

暑さが続く昨今ですが，皆様おかわりありませんか。今月も皆様からいただいたおハガキを集計し，上位10作品を発表してまいります。

さて，「ファイナルファイト」が「GII」こと「グラディウスII」をかわして首位浮上。あっという間に，かなりの得票差をつけるまでになってしまいました。

「ファイナルファイト」は出来のよさも評判ですが，2人同時プレイが面白いという方が多いようです。基本的には協力プレイですが，何かの拍子に相手側のキャラクターにダメージを与えたりすると，もうたいへん。殴り合いにまで発展することもありそうです。アーケードゲームではもったいなくてできませんでしたが。格闘モノとしては意外に簡単な操作性も人気の要因のひとつ。おまけのCDと，エンベロープも評判がよろしいようで。

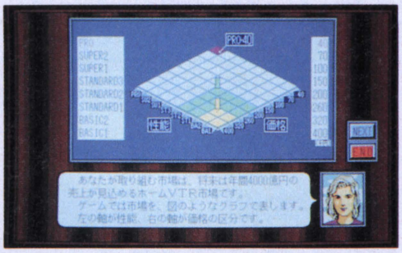
面白くなくなる一方の，実際のF1レースを尻目に（この号が出るころにはもうマンセルがチャンピオンなのでは？），F1ゲームの「OVERTAKE」の人気は加熱気味。画面のセンスには読者からもOKが出ていますから，あとは動きがどこまで磨けるかで人気が決まりそう。

今月は「出たな!! ツインビー」「ジェノサイド2」といったソフトがもちなおしています。原因はわかりません。

「ふしぎの海のナディア」も発売時期が決定してジワジワ人気が出てきました。ひさしくアドベンチャーが上位に来ていないので，ちょっと期待していたいですね。

個人的には「ポピュラスII」の前人気があんまり盛り上がってないのが少し心配ですが，来月の巻き返しを待ちますか。では，今月はこんなところで。しばしさようなら。　(浦)

リーディングカンパニー



光栄からまたまたシミュレーションゲームが発売される。とはいっても，今度出るのはビジネスシミュレーションゲームというやつで，名前は「リーディングカンパニー」。

プレイヤーはもちろん会社社長。そして，その会社は，独自の規格をもつビデオデッキを生産している。ライバル企業は11社，競合する規格は３つあり，このなかで提携を結んだり，妨害工作をしたりしながら，トップの座を獲得しなければならない。

ビジネスゲームとはいえ，特に高度な知識などが要求されることはなく，誰にでもゲームが楽しめるようになっている。

X68000用　3.5/5″2HD版　12,800円(税別)
光栄　☎045(561)6861

(画面写真はPC-9801版のものです)

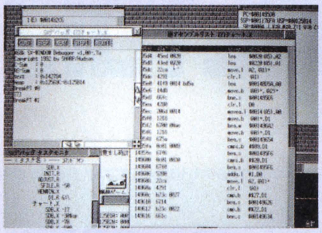
SX-WINDOW開発キット

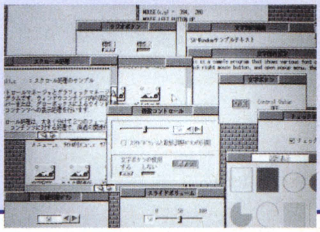
待望のSX-WINDOW開発キットが発売される。内容はSX-WINDOW基本システム一式，SXデバッガ，リソースエディタなどのツール，SX用ライブラリ，サンプルプログラムという構成。これにより，ソースプログラムのエディタから，コンパイル，実行，デバッグまでの一連の処理がすべてウィンドウ上で統合されることになる。

SXデバッガはソースコードデバッグに対応し，マルチウィンドウを生かして，レジスタ内容，スタック，ワークエリアなど，さまざまな情報を一覧することができる。マルチタスクなので，実行中のプログラムのワークをリアルタイムに確認することも可能。リソースエディタでは各種リソースタイプ別に専用のエディタが用意されている。

ディスク５枚組，噂ではマニュアル総量2000ページともいわれ，C compiler PRO-68K以上の大型パッケージになることは間違いない。

X68000用　3.5/5″2HD版5枚組　価格未定
シャープ　☎03(3260)1161





サークII

1990年７月号で紹介した，アクティブロールプレイングゲーム「サーク」の続編が移植される。邪悪な怪物バドゥーを倒し，平和な日々を過ごしていた主人公ラトク・カートが再び旅に出る。今度は母の目を治す薬を探しに。

で，もちろん取りにいって，簡単に手に入るわけはなく，数々のイベントに巻き込まれてしまい，めでたくゲームが成立するのであった。

X68000用　3.5/5″2HD版　価格未定
ブラザー工業(TAKERU)　☎052(824)2493

チェイスH.Q.

“ナンシーより緊急連絡”というセリフで有名なこのゲームは，そのとおりに緊急連絡が入るところから始まる。犯人の車の特徴を確認したら，あとはハイウェイをひたすら走る。目的の車がいたら，ひたすら体当たりをかまし，強引に停車させる。なにしろ，自分が警官なんだから，スピード違反をしようと，犯人の車を壊そうと，おかまいなしなのである。

X68000用　3.5/5″2HD版2枚組　7,800円(税別)
ブラザー工業(TAKERU)　☎052(824)2493

グラディウスIIグッズ当選者発表

グラディウスIIの発売記念として実施された，“I LOVE GRADIUS II キャンペーン”のオリジナルグッズ当選者が決定した。豪華ネームプレートを獲得したのは以下の10名の方々。

(栃木県)片岡誠　鈴木英明　(長野県)加山真一　(東京都)前田哲也　(神奈川県)春日英己　武井真悟　(愛知県)増田光宏　(兵庫県)阿南昌弘　塩路英樹　(佐賀県)山田大作　(敬称略)

なお，2000名に贈られるロゴ入りスポーツタオルは到着するまでのお楽しみとなる。

TREND ANALYSIS

1992年6月の月間売り上げベスト10

POINT	タイトル	発売元	発売日
869	三國志III	光栄	'92/5/28
253	シムアース	イマジニア	'92/5/22
246	太閤立志伝	光栄	'92/5/10
116	グラディウスII	コナミ	'92/2/7
103	スターウォーズ	ビクター音楽産業	'91/12/17
75	ウルティマVI	ポニーキャニオン	'91/6/19
68	出たな!! ツインビー	コナミ	'91/12/6
41	エイリアンシンドローム	電波新聞社	'92/3/25
34	SX-WINDOW ver.2.0	シャープ	'92/3/24
27	レミングス	イマジニア	'92/4/17

やはり，「三國志III」がダントツでトップの座についた。

「三國志III」は，ベストセラーとなった"三國志"シリーズの最新作である。小説や劇画にもなっている中国の古書をもとにゲーム化しているだけに，シリーズを通して，ダイナミックでドラマチックな仕上がりを見せてくれる。

システムもシリーズを重ねるごとに大幅な修正が加えられ，光栄の力の入れ具合を感じさせる。"信長"，そしてこの"三國志"は光栄の大黒柱なのである。

しかし，ジャンルでゲームを区切るのはいいことではないとわかってはいるが，X68000ユーザーの大々的な支持を得るタイプのゲーム，とは少しいいがたいのは事実である。それを跳ね返してのこの結果はかなり立派なことなのではないだろうか。

2位には「シムアース」がズレ込んできたが，数字的には1位とずいぶん差をつけられてしまった。

SX-WINDOW（ver.1.1以上）を持っていて，2Mバイト（あるいはそれ以上）のメモリを持っていなければ遊べないという条件があることはあるが，わりと多くのユーザーがこれを達していると思われるのであまり問題はないはず。もうひとふんばりすることができるだろうか。

で，3位には1位の「三國志III」と同じく光栄の「太閤立志伝」が入って，「シムアース」をサンドイッチ。ここまではシミュレーションゲームが占領している。

4位と5位には「グラディウスII」「スターウォーズ」という老舗どころが腰をすえている。

6位には初登場の「ウルティマVI」が上がってきた。

海外では，「ウルティマVII」が発売されているようだが，人気はいかがなものなのであろうか。

APPLE IIのゲームとして出発した"ウルティマ"シリーズも，いまではIBM PCで真っ先に発売されるようになり，VGAを使用した美しいグラフィックを売り物としている。

また，本編とは別に，システムは同じでキャラクターや舞台設定を変えたシリーズや，「ウルティマ アンダーワールド」という「ダンジョン・マスター」タイプのゲームも発売されている。

新しいものが出ているとはいえ，少し元気がない"ウィザードリィ"に対して，昔からの好敵手"ウルティマ"シリーズは，タイトル数だけを見るとまだまだ元気といったところだろうか。

7位から10位までは，「出たな!! ツインビー」「エイリアンシンドローム」「SX-WINDOW ver.2.0」「レミングス」といったお馴染みのメンバーが続く。POINTを見ていると，全体的に元気がないことが感じられるのは否めない。協力店の皆さんも，ひたすら「ファイナルファイト」に期待しているようだが，はたしてどうなるだろうか。

［データ集計協力店］（順不同）
九十九電機本店
J&P（渋谷/町田）
P&A

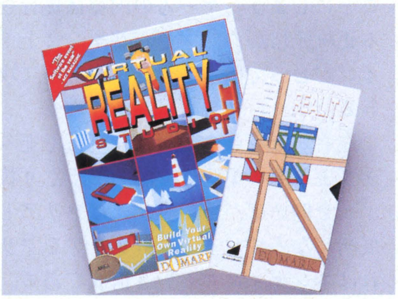
ウワサのソフトウェア（海外編）

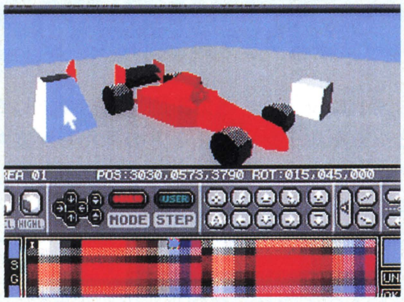
Virtual Reality Studio

また一本取られた。「Virtual Reality Studio」である。なんて素敵なおもちゃだろう。

地面がある。３枚の板をコの字形に置く。これは壁だ。その上に四角錐を置く。これは屋根になる。言葉だとうまく伝わらないのがもどかしい。板や四角錐を「置いた」のである。まず壁の上に屋根を持ってきて，次は下にそろそろと降ろす。壁の上に触れたところで，もうそれより下へはいかなくなる。もちろん，めりこんだりはしない(干渉チェックね)。こうしてできた部屋に入っていく（いわゆるウォークスルー)。奥までくると，コツンと音がして，それ以上前へ進めない。壁につっかえたのだ。

「Virtual Reality Studio」のオブジェクトは本物の積み木以上に積み木である。伸ばしたり縮めたり，積み上げたり色を塗ったり動かしたり。





冷静になって見てみると，干渉チェックもごく簡単なものだし，技術的にものすごいことをやっているというわけではない。もっとも，表示されている３次元の物体をマウスで直接クリックできるというのは，なにげないけれどたいしたものだ。だが，それ以上にすごいのはデザインと割り切りである。厳密さをとりあえず棚上げしてでも，リアルタイムで動作する環境を作り上げた。事実，それらしく動いている。そこがこのプログラムの偉さである。

３次元のモデラと根本的に違うのは，モデリングとレンダリングの区別がないところである。あらゆる操作は即座に目の前の仮想空間に反映する。将来，計算機の性能が鬼のように上がったら，CGシステムはリアルタイムで写実的なレンダリングを行う本物の仮想現実空間になる。

積み木細工して遊ぶだけではない。本来は簡単なアドベンチャーゲームを作るツールだ。物体に属性や条件判定，動きを与えると，「Virtual Reality Studio」の仮想空間はゲーム世界に早変わりする。最近流行のバーチャルリアリティ。データグローブなんてものは使っていないけど，手軽に仮想現実感が楽しめる。

最初に付属のチュートリアルビデオを見た印象は，「面白そうなソフト」。そして実際に遊んでみると，「とても面白いソフト」だった。このビデオは，「このソフトの創造性を制限するのはあなたの想像力だけだ」というような意味の，挑戦的とも受け取れるナレーションで締めくくられている。そう，受け身の姿勢ではこのソフトは使いこなせないといえる。

こういうソフトを作って売ってしまう西洋人にはやはりかなわない。「何かの役に立つんですか？」「全然。でも，面白いじゃないですか」。それでいいのだ。 (A.T.)

発売元　DOMARK（NTSC版）

ウワサのソフトウェア（海外編）

GUY SPY

カナダにREADY SOFTというメーカーがある。この会社はAMIGAの世界では２つの製品，いや，シリーズで有名である。

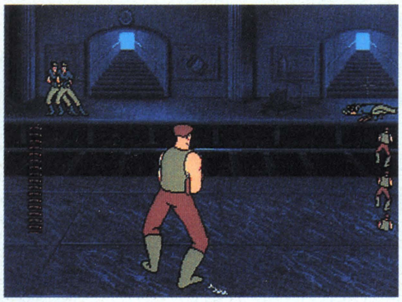
ひとつは“A-MAX”シリーズ。これはソフトウェアだけではなく，パッケージにはハードウェアも含まれている。で，何をするかというと，Macintosh Plusをエミュレートする。Macintoshの128KROMが必要，というところがミソ。

さて，もうひとつのシリーズのほうが今回の主役。これはゲームなのだが，いわゆるフルタイムフルスクリーンアニメーション型アクションゲームとでもいえばいいのだろうか。

ひと昔のゲームセンターに，レーザーディスクを使ったアクションゲームがあったのを覚えているだろうか。画面にはアニメーションが流れていて，その場面に従ってタイミングよく，決められた方向にレバーを入れたり，ボタンを押さなければならないというアレだ。「サンダーストーム」などが有名なのかな。それをパソコン(結構いろいろな機種で発売されている)，しかもフロッピーベースでもできるようにしてしまったというシリーズなのである。

READY SOFTではすでに，中世の騎士が主役の「Dragon's Lair」シリーズを３作，スペースオペラの「Space Ace」を２作，発売している。

しかし，このテのゲームは面白いことは面白いのだが，プレイヤーは決められた入力を探すことのみを要求されるので，アクションゲーム



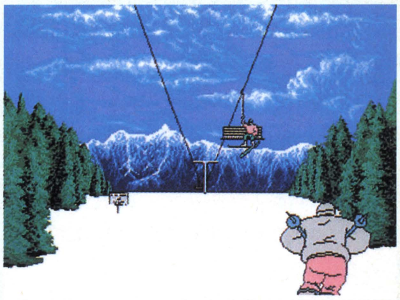
というよりは，記憶ゲームになってしまいがちである。実は人のプレイを見ているほうが面白いということもある。

そこでREADY SOFTも考えた(か，どうかは知らないが)。その結果がこの「GUY SPY」のようなシリーズなのではないだろうか。

まず，アニメーションが流れるのは同じ。しかし，ここではキー入力をする必要はない。ストーリーが進行していくのを見ていればいい。

では，プレイヤーはいつ，何をするかというと，アニメーションの合間に簡単なアクションゲームがいくつも挿入されているのだ。アニメーション場面とのつながりはグラフィック的にも，ストーリー的にもまあまあ自然である。

「オペレーションウルフ」のようなところもあれば，スキーで滑らされるところもあるし，ピラミッドの中をさまよわされることもある。決められたキー入力などはなく，その場で素早く



判断して，操作しなければならない。

全体的な完成度はまだまだという感じだが，いままでの自社製品とは違う，何か新しいものを作ろうという姿勢は見える。人気シリーズにアグラをかいていてはいけないのだ。

発売元　READY SOFT

正義の拳を真っ赤に染めろ

Yaegaki Nachi
八重垣 那智

世紀末の街，メトロシティ。暴力集団マッドギアにさらわれた人質ジェシカを救うために，３人の男たちが立ち上がった。ガイ，コーディー，ハガーが繰り広げるストリートファイト。手強い相手を必殺技でうちのめせ！

誰が決めたのかは定かでないが，男は常に強くなくてはいけないらしい。なぜといわれても困ってしまうが，とにかく強い男というのは，性別を問わず羨望の対象であったことは間違いない。そして，世の中の多くの筋肉ショボショボな人々の願望を満たすために，現実的で科学的なトレーニングとか，楽に筋肉を鍛える怪しい器具や食品，それに非現実的なゲームや映画といったようなさまざまな解決法(？)が生み出されてきた。なかでも誰でも簡単にヒーローになることのできるゲームは，格好の疑似体験として親しまれてきたのである。

熱き血潮の系譜

そういった汗臭い格闘モノを得意とするカプコンが，ついにX68000ゲーム界に参入し，その第１弾として不朽の名作とも呼べる「ファイナルファイト」を移植してきたのである。

またいつもの昔話で恐縮だが，私が初めてこのゲームを発見したときには，まだ，「ストリートファイター'89」という名前であった。それは「テトリス」の大ブームが起こった，1989年の秋のAMショウでのことである。

そのときは，「グラディウスIII」や「R-TYPEII」といった話題作のシューティングゲームが目白押しで，この格闘ゲームにはほとんど注目しなかったので記憶がない。いまから思えば，なぜこのような名作を見逃したのか，はなはだ謎である。

しかし，「ファイナルファイト」として我々の前に姿を現したこのゲームは，格闘ゲームのそれまでの常識を覆すほどの，驚異的な存在感に輝いていた。まさに，この「ファイナルファイト」は格闘ゲームの革命児として，世のプレイヤーを魅了したのである。

素早くスムーズなキャラクターの動き，シンプルな操作，数多くの敵のバリエーション，個性的なボスキャラ，そして絶妙な「難しさ」。どれをとっても新鮮であり，パンチやキックを出すだけで自分が強くなったかのような錯覚に陥る不思議な魔力をもったゲームであったのだ。

このなかでも，素早いキャラクターの操作感というのは特筆すべきことである。連射のかぎりパンチが繰り出され，連続技も自動で鮮やかに決まる。敵に囲まれたら無敵の必殺技を炸裂させれば大逆転も可能だ。それまでの格闘ゲームが，間合いやタイミングを重視して正確にヒットさせなければならなかったのに対し，鬼のような連射で近づいてくる敵を次々と倒せるのである。

このシステムが，いかにエポックメイキングであったかは，これを追従した多くの「もどき」ゲームが，この点をこぞって真似をしていることで，よくわかるのではないかと思う。

闘いの聖書

それでは，具体的にゲームの紹介に入ることにしよう。ゲームの内容は奥行きのある正統派格闘タイプで，基本的に上下のラインが合っているとき，左右のみに攻撃できる。

左右から現れる敵をなぎ倒して進み，さまざまな悪の巣窟での闘いをくぐり抜けてボスを倒せば，そのエリアはクリアという展開になっており，全６エリアをクリアすればエンディングという構成である。

「ガイ」「コーディ」「ハガー」の３人から選ぶプレイヤーは，最大２人同時に闘うことが可能だ。どれも８方向に移動でき，ボタン２つで攻撃とジャンプを行うようにな

忍者，兄貴，市長の３人。誰を選ぶかは君次第

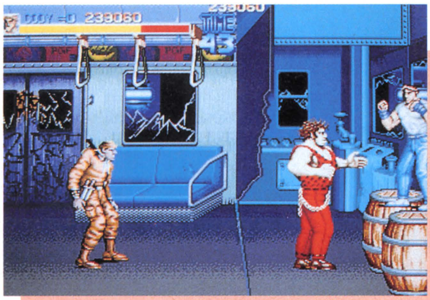
X68000用　5"2HD版2枚組　9,800円(税別)
カプコン　☎03(3340)0718

ガイの特殊技，三角跳びだ

スカウターで相手の体力を見切って闘え

太っちょは結構やっかい。必ず倒さないと

１面ボスのダムンドは高みの見物。口笛で部下を呼ぶ



ちょっとひと休み。でも倒さずには進めない

っている。両方のボタンを同時に押すとそれぞれ違った必殺技が使えるが，一定量の体力を消耗するので乱用しないように注意が必要である。

ゲーム中に敵の攻撃を受けると，画面上に示された自分の体力ゲージが減少していき，すべて赤色になるとプレイヤーをひとり失うことになる。

すべてのプレイヤーを失うとゲームオーバーだが，X68000版では９回までなら継続プレイもできるようになっている。最初はちょっと少なく感じるかもしれないが，逆に多すぎるのも考えものである。継続した回数はスコアの１の位でわかるので，あと何回できるかがプレイ中でも確認できるようになっているのは，ちょっと便利な配慮といえよう。

嵐の前に誓え

このようにオリジナルの緻密さを，踏襲しているX68000版の「ファイナルファイト」であるが，もちろん画面や音といった演出面の移植もしっかりなされている。

画面は384×256のモードにすれば，本物とほとんど同じ画面比で楽しめるし，ちょっとクセのあるサウンドも忠実な移植だ。逆にPCM音などは，X68000版のほうがクリアな感じを受けた。MIDIにも対応しているので，まずまずの合格点といったところだろうか。オリジナルのサウンドは，特別付録の「スペシャルCD」にも入っているので，聴き比べてみることをお勧めする。

ゲームの設定変更も，タイトルからオプションモードに入ることで簡単にできるようになっている。難易度やプレイヤー数なども変えられるので，幅広く遊ぶことができる。ちなみにデフォルトでセットされているデータが，オリジナルの標準なので，いかにオリジナルが厳しかったか，ちょっと体験してみることもいいだろう。

ほかにもMIDIの選択，画面モードの切り替えや，ジョイスティックとキーボードの切り替えなどもこのモードで行う。さらにコンフィギュレーションモードに入ることで，各操作の割り当てを変更することも可能になっているのはありがたい。

また，今回の「ファイナルファイト」はハードディスクにインストールすることができることも特徴に挙げていいだろう。多少OSの知識が必要だが，基礎的な知識のレベルでもそんなに難しいようには感じなかったので大丈夫だと思われる。

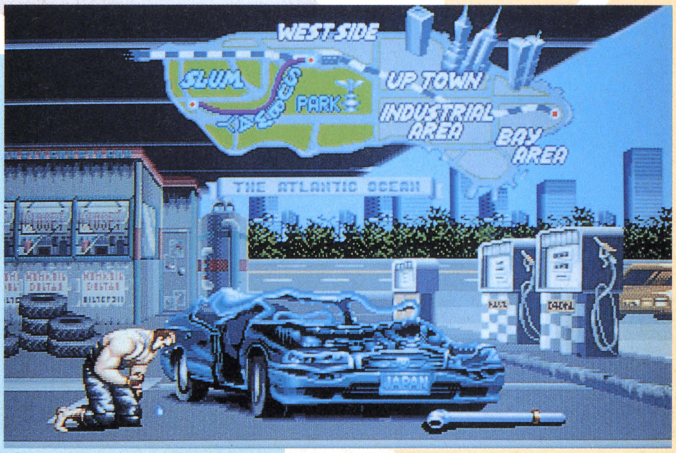
あと，ちょっと気になることは「要２Mバイト」のソフトであることだろうか。

だからといって完全オンメモリなわけでもなく，一部ではエリアの途中でアクセスしたりもするのはいただけない。そういったところには，もう少し配慮がほしかったところであろう。

自信を闘志に変えろ

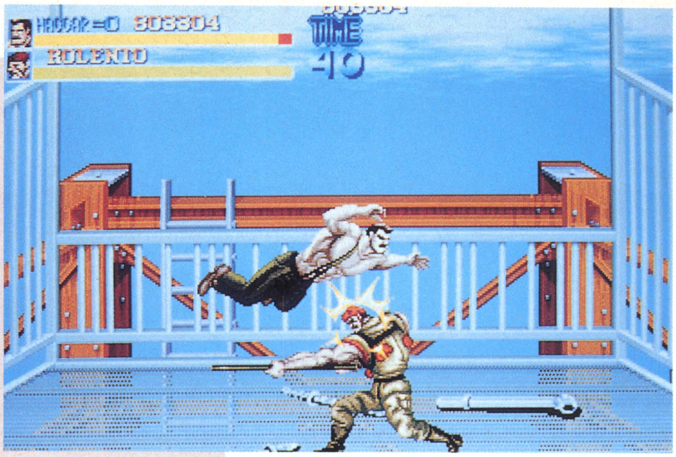
それではゲームを始めてみよう。いちば

理不尽に強いソドム。無敵のタックルには注意

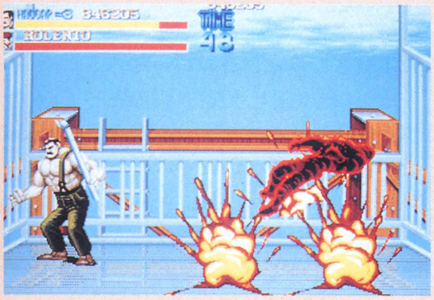


ボーナスステージで車を壊されたかわいそうなお兄ちゃん

悪徳警官エディの拳銃乱射６連発にはかなわない



レッドベレーのロレント。棒の直撃をくらうと痛い



２つ目のボーナスステージはガラス板割り

ん最初はキャラクターの選択である。まあ100パーセント趣味で選んでもかまわないのだが，初心者は誰を選べばいいのかわからないかもしれない。各キャラクターの紹介を兼ねて，その問題に答えてみよう。

●ガイ

忍者の血を引く格闘家で日本人という設定は，この際どうでもいいだろう。スピードに長け，機敏な動きからの連続技が得意である。動きや技の使い分けを覚えるには適しているかもしれない。ただパワーが弱いので，後半はかなりきついことが予想される。独自の技に壁を使った「三角飛び蹴り」がある。

●コーディ

マーシャルアーツの達人で，ナイフを持って連続攻撃ができるのが特徴。スピードやパワーのバランスも取れており，きっちりと攻略すれば確実なプレイが見込める。ナイフの使い方がとにかくカギになる。

●ハガー

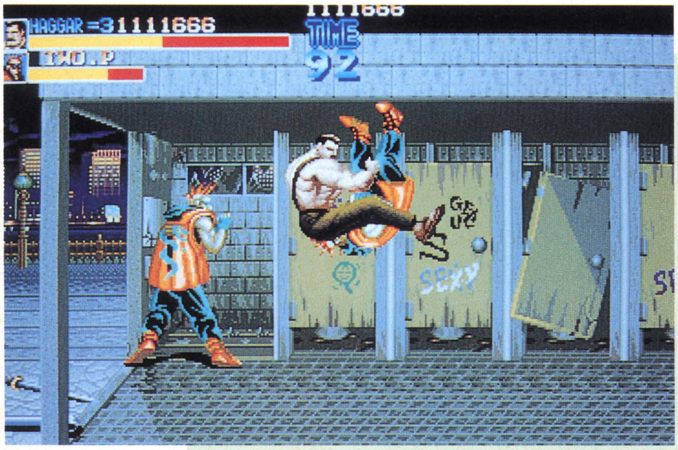
パワーは随一だがスピードに欠けるのが難点。ほかの２人とは使い勝手が異なるかもしれない。独自の技にパイルドライバーがあるが，どちらかというと連続パンチからのバックドロップのほうが利用価値は高い。なぜなら，バックドロップ中は無敵だからである。敵を連続で殴りながらバックドロップをかければ，集団の敵に挟まれた場合でも有利に戦えるのは大きな利点だろう。

こうしてみると別に初心者だから，誰がいいということではなく，重要なのはそのキャラクターを使いこなすことであろう。説明書には細かい技の出し方が出ていないので，蛇足ながらいくつかの有用な技を説明しておくことにする。

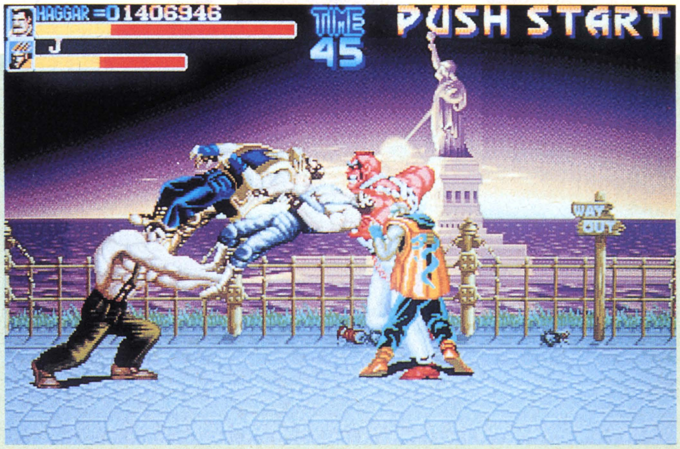
パンチからレバーを上か下に入れておくと「投げ」，敵のそばで敵のほうにレバーを入れると「つかみ」，つかみからレバーを入れずに攻撃すれば「折檻(せっかん)」，さらにレバーを入れて攻撃すれば「投げ」，ジャンプ中に下レバー＋攻撃で「プレス」，という感じである。ちょっとマスターするのに手間取るかもしれないが，１つひとつ確実にモノにしていけば，しだいに先に進める自分に気づくだろう。

力尽きても闘え

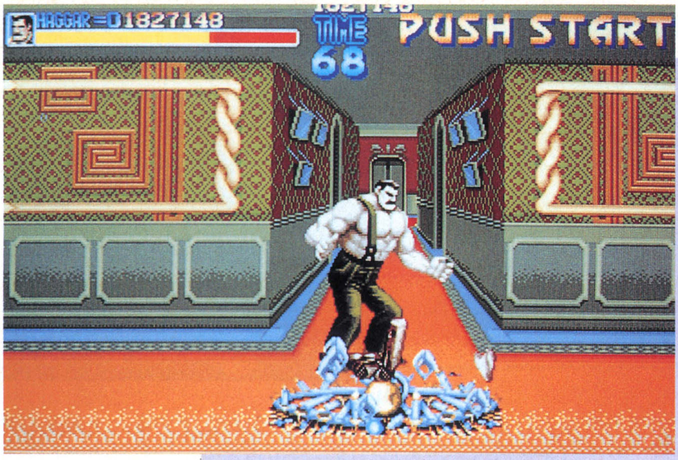
ここでやっと具体的な面の紹介だが，いつもどおりあまり詳しくは述べないで，簡単にすませることにする。具体的な攻略の



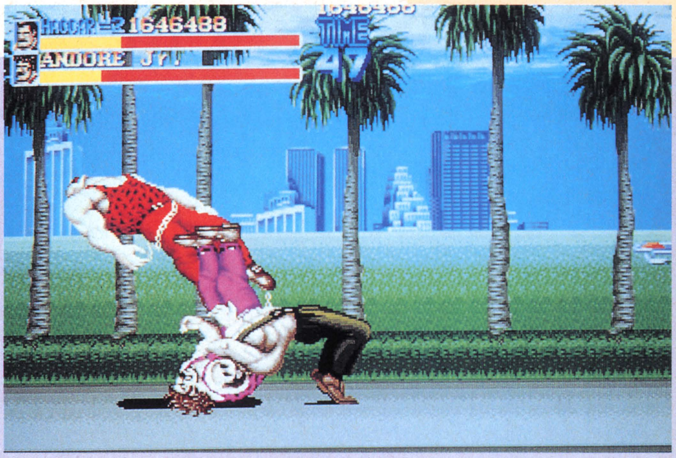
威力バツグンのパイルドライバー。スクリューはしないけど



ゆでダコのようなアビゲイル。投げをくらうと涙が出るぞ



シャンデリアは落とせる。ボーナスはしっかりいただこう



屋上庭園での死闘。ボスまではまだ遠い

要素が少ないことは勘弁してもらいたい。

エリア1　スラム

まさに街のゴミ溜め，スラムからゲームはスタートする。全体的にそうだが，敵が画面外から突撃してくる場所は，一度痛い目にあったら必ず忘れないことである。皮ジャンと軍服を着た奴は，後々ライバルになるので，動きを見切っておきたい。

エリア2　地下鉄

地下鉄に乗る前は，屈指の猛攻にさらされるので，敵の出現する場所を覚えておこう。ボスのソドムは，真上や真下にはタックルできないことをうまく利用すること。この面までノーミスで抜けられれば，かなりの実力が備わったと思っていいだろう。

エリア3　ウエストサイド

酒場から地下プロレスを経て，悪徳警官のボスと対決する。警官が吐き捨てたガムは必ず拾うこと。追い詰めて一気に倒さないと，銃を乱射されて悲惨なことになるので注意しよう。

ドラム缶を壊せばダイヤも出る

最後のボスとついに対決

エリア4　工場

1Pのプレイヤーは最初に上下に動かなければ，3本目の噴射炎からは安全地帯になっている。エレベータでは隅で闘っていればいいが，ボスのロレントの卑怯な攻撃には，てこずること必至。

エリア5　港

なかなか厳しいエリアである。敵を小出しにしていると時間がなくなるので，落ちているアイテムを利用して，無駄な時間を減らすようにしよう。ボスのアビゲイルは登場のときに先制攻撃をかけて，少しでも有利に闘うといい。

エリア6　アップタウン

随所で敵の集中攻撃がある。特に軍人のナイフ攻撃には，細心の注意を払いたい。屋上に上がって建物に入り直してからが，本当の最終面といえるかもしれない。最後のボスを倒し，感動のエンディングを見ることがはたしてできるのだろうか？

繊細な格闘家

とまあ，ざっと眺めた程度では，この移植の出来は高いといってもいいだろう。操作感覚は再現されているし，画面の印象も忠実である。しかし，やはり移植であるがゆえの問題点がないわけではない。ケチをつけるようだが，気になったので書いておこう。

ひとつはキャラクターの数の制限である。画面に敵が最大4人までしか登場しないことで，オリジナルで培ったパターンが通用しないことがあるのは，残念なことである。出てこれなかった敵は，途中で空きができると現れたりするので，ちょっと調子が狂ってしまう感じがした。また道中の樽や障害物も若干少ないので，なにか損をした気分になったのだが，どんなものだろうか。

もうひとつは効果音の問題である。別な音が鳴ると前の音が消されてしまうので，無音でボスにタックルされたり，音もなくドラム缶に轢かれたりするのは，ちょっといただけない。特定の場面である種の効果音を制限して，納得いくようにしてほしかったが，これは贅沢な悩みなのかもしれない。

こういった細かい不満はあるが，全体的に見れば，本格的な格闘ゲームがついにX68000に移植されたということで，高く評価できるといえるだろう。非常に奥の深いゲームなので，買ったからにはぜひ極めてほしい。近代格闘ゲームの，いまだ超えられていない頂点であり，究極である世界を，堪能してほしいと思うのだ。

俺より強い奴に会いにいく

この原稿を書いていて，カプコンの特徴ってなんだろうと考えてみたが，カプコンのゲームに共通する要素に，自分ではあまり気づくことはできなかった。思い余って，マニアの伊謡見氏に聞いてみると，「100万点以上で残機が増えない」という意味不明の答えが返ってきた。出るゲームの半分以上が体力制だといっても，あくまで主張するので，カプコンの特徴は入れ込んだ筋金入りのマニアがいっぱいいることだと気がついた。世の中って深いなぁ，うんうん。

総合評価	0　　5　　10
ゲーム性	★★★★★★★★
技術	★★★★★★★
サウンド	★★★★★★
グラフィック	★★★★★★★★
爽快感	★★★★★★★★★
Oh! My God!	★★★★★★★★★★

平家討伐道中膝栗毛

Kageyama Hiroaki
影山 裕昭

このゲームは，シネマウェア社で発売されていた「Lord of the Rising Sun」をバージョンアップして，日本向けにアレンジしたものだそうだ。基本システムはアメリカ生まれというわけだが，はたしてどんなもんだろうか。

この「ライジングサン」は，プレイヤーが源氏一族の大将となって平家を倒し，日本のすべての城を占領して全国統一を目指す，リアルタイムシミュレーションウォーゲームである。

プレイヤーは源氏一族の長を演じるわけだが，最初に源頼朝か源義経かの選択を迫られる。ご存じのとおり頼朝は鎌倉幕府を開いた人で，義経（幼名牛若丸）はその弟。壇の浦で平家を滅ぼしたことで知られる。

どちらを選ぶのも自由だが，どうせなら義経で遊んでみたい。義経は自害こそしたものの，頼朝に殺されたようなものだからだ。あなたが源平討魔伝を遊んでいたなら，ヒョーヒョー叫ぶ義経を殺して幾度となくギョエ～と叫ばせていたに違いない。いまこそ義経にいい思いをさせてあげるのだ。義経となり全国統一前に頼朝を殺してしまえ。義経も浮かばれる。世は戦国，身内同士の殺し合いもめずらしくない時代だ。

ザッザッザッ，移動だ

このゲームではターンの概念はなく，基本的にリアルタイムでゲームが進行する。ゲームを始めると，画面上に全国マップの一部が表示される。これが戦略画面であり，「ライジングサン」の基本となる画面である。操作はフルマウスオペレーション。スクロールさせて，九州から東北までを戦略画面上で見ることができる。

戦略画面では19の城，都市や寺のある場所(これらを拠点と呼ぶ)，および武将のいる位置を確認することができる。武将の移動はすべて戦略画面で指示をする。移動は城から城，都市から城，といったように拠点から拠点への移動である。

指示方法は移動させたい武将の上にカーソルを合わせて左クリック。するとカーソルの下の武将が白く光る。ドラッグしたまま移動先にカーソルを持っていき，そこが移動先として認められればカーソルの下が白く光るので，マウスから手を離せばすぐさま移動を始めるようになっている。街道に沿ってトコトコと移動する姿は愉快だ。

しかし，いったん，指示を与えたあとは目的地に達するまで行き先を変更することができなくなる。これは結構きつい仕様だがしかたがない。

パラメータ地獄はない

パラメータ地獄なんて言葉を使ったが，歴史シミュレーションものというと，パラメータがズラズラッと並んでいるという印象が強い。しかし，「ライジングサン」では数値化されたパラメータはひとつもない。プレイヤーが知ることのできるパラメータは武将の兵力と士気の2つだけ。

これは戦略画面で，調べたい武将の上でマウスを右クリックすると，ウィンドウに棒グラフで表示されるようになっている。ちなみに武将の移動を実行すると距離に比例して兵士の士気が低下する。士気を高めるには城，都市や寺といった拠点で何もしないでジーッとしていればいい。

戦略画面上では見れないが，武将の技量レベルなどを表示するスキルウィンドウというものもある。ここでは武将の統率力，剣術などが0～6の範囲で表される。統率力なら軍配の数，剣術なら剣の数といったふうに，数字ではなくビジュアルで表示されるあたりは，実に親しみやすい設計になっている。

ウォウォウォ，遭遇だ

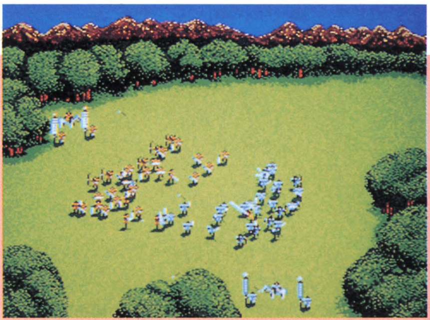
移動途中に，源氏一族以外の武将と出会うと遭遇イベントが発生する。画面は戦略画面から遭遇画面に切り替わり，合戦を決意するか，回避するか，連合を組むか，といったなかからコマンドを選択する。ここではリアルタイムではなく，ゆっくりとコマンドを選択することができる。

争いを避けようとしても，必ずしも避けられない場合もある。もし合戦を決意すれば，そのまま合戦画面に移行する。

また，配下の武将が移動中にほかの陣営と遭遇した場合は，画面下にその旨メッセージが表示される。このとき，自分で選択することもできるが，選択しなければ配下の武将の判断で合戦の決意，回避などの行動が自動的に決定する。勝手に決められては困る，という場合以外は，わざわざ選択するために遭遇画面に移行させることもな

X68000用　5"2HD版　9,800円(税別)
ビクター音楽産業　☎03(3423)7901

戦略画面ではほかの武将の行動が表示される



おもちゃの兵隊を見ているような合戦

い。手間がかかるだけだ。配下の武将の判断で合戦になったときは，戦力や武将の技量を比較したうえ，戦闘結果が画面下に表示されるようになっている。

カキンカキンカキン,合戦だ

合戦を決意すると画面は合戦画面へと移行する。ここでは兵力に応じて，画面左上にプレイヤーの兵士，右下に相手の兵士が表示される。マウスを左クリックすると戦闘開始の合図を知らす角笛が響き渡り，両軍の兵士が一斉に敵陣目がけて移動を開始。突撃じゃ〜。

ちっちゃな兵士が刀を振り回して移動するさまは，緊迫感があるどころか，見ていて楽しい。画面中央で刀が交わり，刀と刀がぶつかり合う音が鳴り響く。ここでプレイヤーができるのは，カーソルで兵士もしくは弓兵の移動する方向を決めることだけ。

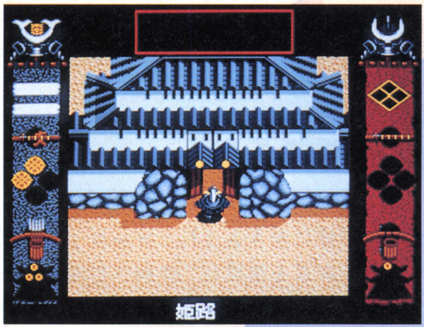
合戦の結果，敵を倒し，そのときにカーソルを馬に乗った敵大将に合わせて左クリックしていると馬追いに入ることができる。

馬追いは斜めスクロールのアクションゲームである。白馬に乗った自軍の大将をマウスで上下左右に操り，障害物を避けながら敵大将を追っていく。マウスの両クリックでは，太刀を振って敵兵を切り倒せる。斬鉄剣ではないので間違っても松は切り倒せない。運よく敵大将と出会い，見事切り倒せば敵大将を殺すことができる。気分はもう暴れん坊将軍である（ウソ)。

ピュンピュンピュン,攻城戦,防城戦だ

城を攻めて落城させることと，奪った城を敵の攻撃から守ることが「ライジングサン」での最重要事項である。自分の領土はなく領土間の境界線もない。どんな武将でも九州から東北まで自由に移動することができるのである。

こちらが城を奪い取る場合，その城に武将がいなければ無条件に城を占領することができるが，敵がいれば攻城戦になる。攻城戦は制限時間つきの迷路ゲームの感覚だ。



攻城戦はひとりで敵の城に乗り込む

敵の城の大手門から侵入し，入り組んだ内部を主殿を目指して進んでいく。４方向スクロールで先が見えないから，どこが主殿なのかはわからない。袋小路もある。ときどき現れる敵兵を斬り倒すと制限時間が増え，やられると制限時間が減る。いちおうアクションゲームのノリであるが，そんなにむずかしいことはない。

なんとなくアクションRPGのノリで，やや現実性に欠けている気もする。出てくる敵兵の数も少ないし，どうせならチャンバラのようなノリがほしかったなあ。

逆にむずかしいのが防城戦である。防城戦は城に入り込んでくる敵兵を弓で射るアクションゲームである。敵はそんなに素早く動いているわけではないのだが，敵に矢を当てることはむずかしい。おそらく，お祭りなんかにあるライフルでヌイグルミを倒すゲームよりむずかしい。城に関しては「攻めるは易し，守るは難し」だ。

シュッシュッシュッ,忍者だ

私の名前を見て，祖先は忍者だという人がたまにいるが，さてどうなんでしょう。そんなことをいう人は忍者赤影のファンだったに違いない。「ライジングサン」では忍者を使って，対立する武将を暗殺する作戦もとれる。しかし，その成功率は低く，失敗すればハラキリがあなたを待っている。当然ゲームオーバーである。忍者はなるべく使わないほうがいいだろう。

狙いどおりに当てるのは至難のワザ

しかし，こちらが使わなくても，相手が忍者を送り込んでくることはある。ゲーム中に突然，戦略画面がフェードアウトしたら，その合図。ここでもバリバリのアクションゲームが待っている。マウスで刀を動かして，忍者の投げてくる手裏剣を弾き落とすのだ。マジで手に汗握る緊張感が味わえる。なんといっても突然だし，手裏剣をいくつか体に受ければ殺されてしまうのだ。

敵の手裏剣を一定数弾き落とすと，味方の兵が駆けつけて忍者を倒してくれる。味方が忍者を倒せば，忍者を送り込んできた敵大将がハラキリである。「ライジングサン」のアクションシーンのなかでいちばん盛り上がるといってもいいね。

まとめてみると

結局「ライジングサン」というゲームは，リアルタイムシミュレーションにアクションゲームをミックスして，両方の美味しい部分をプレイヤーに味わってもらおうという志の高いゲームを目指したものである。

ゲーム開始直後から次々といろんな武将に攻められて，じっくり考えている時間はないが，退屈せずに楽しめていい。後半になって源氏と平家の２大勢力になると，城の取り合いが激しくなってゲームがますます面白くなってくる。城が集中攻撃を浴びてるときなんてのはドキドキもんだよ。

自室でくつろぐ

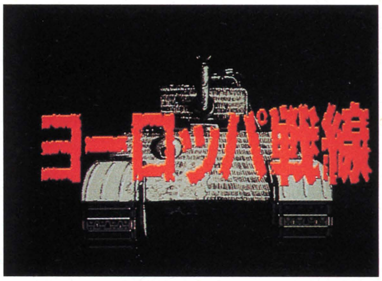
X68000を生かしています

X68000ではメモリを２Mバイト実装しているユーザーが増えてきた。そんなわけで，ゲームソフトも大容量メモリに対応してきている。メモリ上に一括ロードしてディスクアクセス回数を減らすとか，ディスク交換を減らすとかね。「ライジングサン」では２Mバイト以上のRAMを積んでいるX68000で遊んだ場合は，ディスク交換の手間が省略でき快適である。しかし，欲をいえばハードディスクにインストールできるようにしてほしい。

また馬の嘶き(いなな)きをはじめ，随所でAD PCMによる効果音が使用され，臨場感を高めるのにひと役買っている。マウス標準装備ならではのフルマウスオペレーションもうれしい。マウスによる操作は快適で操作性はいい部類に入る。

総合評価	0　5　10
戦略性	★★★★★★★★
操作性	★★★★★★★★
グラフィック	★★★★★★★★
効果音	★★★★★★★★
忍者の襲来	★★★★★★★★★

欧州戦線異状なーし



第２次世界大戦の戦禍に巻き込まれたヨーロッパを舞台に，連合軍，あるいは枢軸軍を指揮し，勝利へと導く。そんなシミュレーションゲームが光栄から発売された。人間的要素が重視されているのが，光栄らしいところかな。

Kaneko Shunichi
金子 俊一

第２次世界大戦を題材にしたシミュレーションゲームがまたもや登場した。光栄としては「提督の決断」に続くもので，WWIIゲームと呼ぶそうな。

太平洋が舞台だった前回に比べ，今回はヨーロッパ大陸が舞台となり，戦車を操るのがメインとなった。同様のテーマを扱ったソフトとしてはシステムソフトの「ブリッツクリーク」がある。

私は軍団長

プレイヤーは４個師団を任されている軍団長である。軍団長の部下は６将軍。その下に作戦参謀，情報参謀に師団長が４人というぐあい。

このゲームには将軍が10人ぐらいいて，その中から軍団長，参謀などを選ぶことになる。それぞれに光栄お得意のパラメータがあって，指揮能力，魅力，体力などが並んでいる。

１個師団には４個連隊＋２個戦闘団＋１個工兵団＋１個補給団の計８部隊が編成される。連隊/戦闘団は戦闘を目的とした部隊で，連隊には６個大隊，戦闘団には４個大隊まで配備できる。工兵団は地雷の設置や橋を架けるなどの工作を担当し，４個大隊まで配備できる。補給団は補給隊と修理隊を合わせて４個大隊まで配備できる。

地図上では大隊は表示されずに，連隊や工兵団などのひとまとまりで表示される。

つまるところ，軍団長を頂点にしたツリー構造の軍団が浮かび上がってくる。その数たるや，最大で160大隊ということになるのだ。

補給関係では，補給ルートを断たれると補給が受けられなくなっている。持久戦も考慮した戦いができるようになったわけだ。

工兵団やら修理隊などは，なかなか憎いところに目をつけたと思う。橋がなければ架ければいいし，敵に有利な橋なら壊せばいい。戦場では当たり前だよね。また，壊れた兵器は捨てるばかりがすべてではない。思いだすなあ，マチルダさんを。

作戦を命ずる

作戦は年代ごとに６つあり，枢軸軍（ドイツ）か連合軍（イギリス&フランス）を選んで戦うことになる。２人プレイもあるので，友達と争うこともできる。また，この６つの作戦を通して行うシナリオモードというやつもあるが，シナリオモードでは枢軸軍しか選べない。

それぞれの作戦は軍上層部から命令を受けるかたちになる。都市占領か都市防衛が作戦のすべて。中間管理職はつらいやね。

しかし，いくらなんでも160もの大隊にいちいち指示を出すというのは現実離れした行動である。ってことで，面倒だったら師団ごと師団長に任せてしまえる。ひとつの師団に４つまで作戦を伝えられるが，作戦に優先順位がないのはちょっと使えない。また，予想されることだが，結構おバカなので，あまり期待はしないほうがよい。

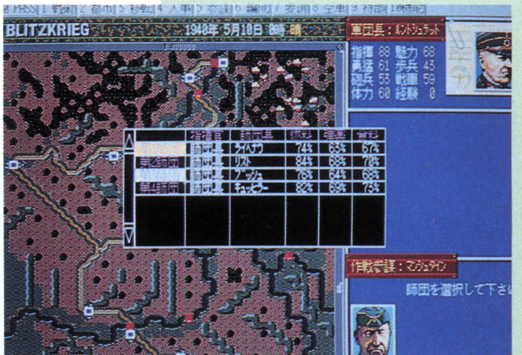
別に面倒臭くないなら自分ですべて指示するのも一興なのだろうが，なにせ160大隊ですぜ，旦那。実際の戦場だってそんなことはやらないよね。バカな部下をもった上官の気持ちも体験できると考えたほうがよさそうだね，こりゃ。

ところで，似たようなシステムを取っている（元祖？）「銀河英雄伝説」では，独立部隊というものが存在していた。どの将軍にも属さない部隊なので，プレーヤーが指示する必要があったが，このヨーロッパ戦線ではすべての大隊はどこかの連隊やら補給団やらに入ってしまう。つまり，どこかの師団に所属することになる。４つの師団をすべて師団長に任せてしまうと，プレーヤーは待つだけになってしまうのだ。はっきりいってヒマ。

任せ方も時間単位（１～24時間）で全面委任というかたちになる。途中で指示を変えたくても中断はできない。任せるといった以上，将軍に二言はないのである。

ついでにいわせてもらえば，時間という概念を導入したのならば，すべての大隊が同時に移動/攻撃してほしいし，補給中に敵の攻撃があったら，補給が中断されるとか

作戦ファイルふうのシナリオ説明



選択はポップアップメニューっぽい

X68000用 3.5/5"2HD版3枚組 12,800円(税別)
光栄　☎045(561)6861

のリアルさがほしい。いまだに各部隊ごとに移動などがシリアルに行われるのは失格だと思う。ボードゲームのレベルを抜けていないではないか。あーダサダサ。

戦闘シーンはクォータービュー

あれ？　まるで先月の「三國志III」のタイトルのようじゃないか。そう，この「ヨーロッパ戦線」でも戦闘シーンにクォータービューを採用したのだ。

しかし，このクォータービューも普通のHEX戦のようなもの。見た目がよくなっているだけである。せっかくクォータービューを採用したのだから，いままでにはなかったような演出もほしくなる。たとえば，戦車の流れ弾によって建物が破壊されるとか，火事が発生して町が焼けてしまうなどである。不可能とはいわせないぞ。天下のコウ・シブサワなんだから，そのくらい思いつかないのかなあ。

文句はそのくらいにして，この戦闘シーンでは10ターン以内なら何度でも命令が出せる。部隊によっては1ターンで何度も攻撃できるのもいれば，準備不足で攻撃できないものすらいる。これではターンという概念には当てはまらないのではないだろうか。ここでも時間を導入すればよかったのにね。

師団長に任せた部隊だと，自分で指示できないので，ボーっとクォータービューを眺めることになるが，これまたエライ時間がかかる。結局，戦闘は見ないモードというやつで遊ぶのが快適だったりして，なかなか意味なしなんだな，これが。

移植に関して

X68000のマーケットなんて国民機に比べれば微々たるものなのかもしれない。しかし，10万人のユーザーがいるいまとなっては，ソフトを出してもらえるだけで素直に喜ぶような奇特な人は少なくなっているはずだ。

光栄のゲームは国民機からX68000に移植される場合がほとんどだろう。完全なベタ移植ではないとはいえ，ユーザーを納得させるような移植ができているかといえば，疑問符を出さざるをえない状況である。このことは，ゲームを買った人ならば誰しもが認める点であろう。

そこで，X68000ユーザーとして希望などを述べてみたい。

まず最低限の事項として，ディスクの管理が挙げられる。X68000にはオートイジェクトの機能がついている。まさか光栄の開発の人が知らないことはないだろう。また，プログラムでイジェクトさせることが簡単なのも有名だ。ってことで，ディスクの入れ替えのときはイジェクトしてほしい。いや，イジェクトしなければならない。

また，X68000のディスクドライブがインテリジェント方式ということは周知の事実。ディスク挿入の認識くらいはオチャノコサイサイなのである。ディスクを入れ替えたあとにクリックさせたり，リターンキーを叩かせるのは邪道以外の何者でもない。

また，X68000には広大無比なグラフィックRAMが搭載されている。国民機との兼ね合いで，テキストRAMの16色しか使わないのは我慢しよう。しかし，グラフィックRAMを遊ばせておくのは許せない。このゲームでもほとんど使用していないことを確認した。キャッシュメモリにでもすればいいのに。無理ならRAMディスクでも結構。メモリの増設に対しても同様。細かいところでのディスクアクセスが多すぎる。

以上のことが無謀な注文かどうかは，ほかに市販されているX68000用のソフトを見れば一目瞭然。いろいろな制約のなかで移植しているのだろうが，これらは最低限のたしなみで，技術力や開発時間/費用とは関係ないレベルの話だと思う。もちろん，ほかのソフトハウスも肝に銘じてほしい事項である。こういったところで手を抜くと，不思議とほかのアラが目立ってくるから，とっても損でしょ。

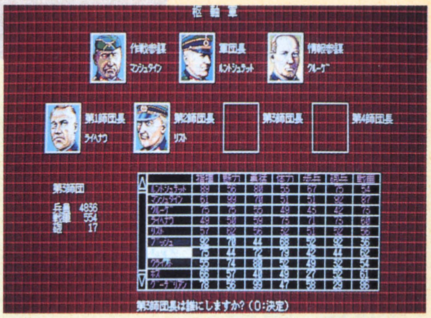
禁断のまとめ

もし，あなたが熱烈な光栄の信奉者で，我慢強くて，戦車系のシミュレーションゲームをやりたいというならこのゲームは勧められる。この3つの条件をひとつでも外している人では，たぶん納得いかないのではないだろうか。

シミュレーションゲームである以上，ある程度の我慢強さは必要である。よって残る2点に着目すると，戦車系のシミュレーションでなくてもいいのなら，「三國志III」を買ったほうがいい（他社製品じゃないから許してね)。光栄の信奉者でなければ，「ブリッツクリーク」のほうがよくできていると断言する。

光栄からは，諸手をあげて喜べるような，もっと素晴らしいゲームと出合えることを期待したい。それを作るだけの力を持ったソフトハウスだと私は信じている。

町を占領。お決まりのグラフィック


将軍を選ぶところ。ロンメルもいる

これがウワサのクォータービュー

ほかにもいわせて

大地図と中地図という2種類の地図があるが，大地図では細かすぎて情報が得られないし，中地図のスクロールは精神衛生上よろしくない。また，いつでもスクロールできるわけでもない。せめてスクロールバーくらいあれば。

今回の音楽担当者は，「ルパン三世」などでお馴染みの大野雄二氏である。坂本教授といい，大野雄二氏といい，超豪華なキャストは光栄の特権だろうか。次は久石譲氏だったりして。

で，肝心の音楽はというと，どうも音楽性の高さを音色で壊しているようなので，ぜひともMIDI対応にしていただきたい。FM音源だって頑張り方しだいで結構いい音が出るんだけどね。とりあえず，X68000はMIDIの普及率も高いのであるから。

総合評価（0　5　10）

システム	★★★★★★
グラフィック	★★★★★★★
ミュージック	★★★★★★
サウンドエフェクト	★★★★★★★
ディスクイジェクト	★

燃えるツールで輝く星々

Ishibumi Akira
伊溢見 あきら

少し前に，アモルファスの「シューティング68K」を使った作品のコンテストが行われた。そして，その受賞作がTAKERUで発売される。グランプリは単独で，優秀作は2本セットとなっているけど，どっちを買うのが得なのかなあ。

ゲームを遊んだことがあるなら，誰しも必ず，こんなゲームで遊びたいなぁというアイデアのひとつやふたつはもっているはずだ。それを実現できるかどうかはともかく，夢は大きいほうがいい。

最近面白いゲームが少ないと嘆く私なども，そんな夢を集めたコンテストが行われるとなれば，燃え上がらせてくれそうなゲームが現れるのではないかと期待する，今日この頃である。

作れや作れのコンテスト

さて，「シューティング68K」というソフトを知っているだろうか。電源オンで即起動，マウスひとつで楽々ゲームが作れてしまう，縦スクロールシューティングゲームのコンストラクションツールだ。本誌では昨年(1991年)の7月号で，横内氏による克明なレビューがなされているので，忘れてしまったり，知らなかった人はそちらも参照するといいだろう。

結局これはツールであるから，買ってモノを作るのが身上ということになる。だが，モノを作るということは，並々ならぬ根気と努力が必要であることは，あえて書くまでもないことだ。

ヴァリストレスナルトの1面。まだ簡単

ヴァリストレスナルト（グランプリ作品）
三國志ー幻伝ー，フェイバー・ザ・ロード(優秀賞)
X68000用 3.5/5″2HD版2枚組 各3,000円(税込)
ブラザー工業(TAKERU) ☎052(824)2493

そこで，そんな努力と忍耐の備わった全国の「シューティング68K」ファンのための，ファンによる自作データのコンテストが行われたのである。

多数の応募作のなかから厳正な抽選，じゃなくて審査が行われ，栄えあるグランプリ1作と優秀賞2作が選ばれた。そして，これらをそのまま埋もれさせてしまうのはもったいないと，この3作品がこのほどブラザー工業のTAKERUで発売されることが決定した。

グランプリは単独で，優秀作は2本セットでのパッケージとなる。もちろん，これらは「シューティング68K」を必要とせず，単独で遊ぶことが可能になっている。では，さっそくそれぞれの解説をしていこう。

さすがの貫禄，グランプリ

まずはグランプリの紹介からいこう。題名は「ヴァリストレスナルト」という作品だ。自機が戦闘機という，最もオーソドックスなタイプの仕上がりを見せている。

「シューティング68K」というツールは，基本的にゲームのシステムが変更できないので，そんなに鋭いモノではないだろうと，半分も期待しないで起動してみたのだが，正直いってこれにはかなり驚かされた。思わずグランプリに納得してしまったほどの完成度を，この作品はもっているといえるだろう。

すべての作品に共通する，基本的なゲームの内容を最初に簡潔に説明しておくことにする。原則的には空中を飛ぶ自機で戦う，縦スクロールのシューティングゲームである。各面の最後に登場するボスを倒せばクリアとなり，最終面をクリアすればめでたくエンディングとなる仕組みだ。パワーアップは，特定のアイテムで特定の機能が得られるタイプしか選べず，ミスをしてもその場からゲームが継続するというシステムも固定だ。

そんな制約のなかで，「ヴァリストレスナルト」はいかにグランプリたるべき作品になりえたのかを細かく見ていくことにしよう。

最初に気づいたのは，その背景の美しさである。色彩感覚も含めて，ここまで緻密に書き込む努力は，想像するだけでもたいへんだったろうと思われる。全体的に「出たな!! ツインビー」のようなイメージがあるが，丸写しというわけではなく，独自の感覚として処理されているので気持ちがいい。遠慮せずにいろいろなものからオイシイところを盗むのも(人聞きは悪いが)，いい作品を作るうえでは，意外にポイントになるのかもしれない。スプライトのキャラクターも，水準以上の立派なレベルを保っていて，全体的な画面の印象は，非常にきれいな仕上がりだ。

力作のテクを見切れ

2番目に挙げるのは，音楽がかなり凝っているということである。全曲オリジナルで構成されているのはもちろん，さらには，システムの弱い部分をカバーする工夫も見受けられる。

「シューティング68K」では，背景スクロールの設定によっては，OPMファイルの演奏のテンポが遅くなるという症状が現れるときがある。それはこのゲームにおいても

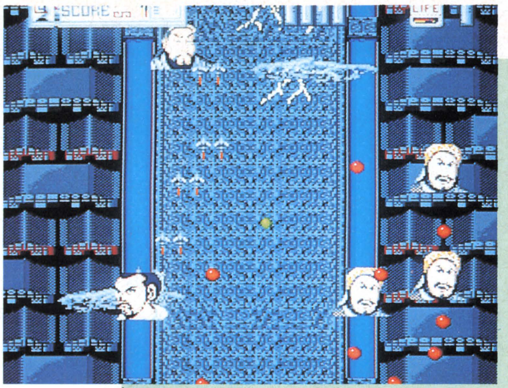
ハッとさせられる美しさ。質感に注目

例外ではないので，この作品ではそれを見越してBGMのテンポを速めに設定することで，それを回避している。休符で演奏開始のタイミングを合わせてみたり，いろいろと細かいところにまで気が配られているのにも注意したい。

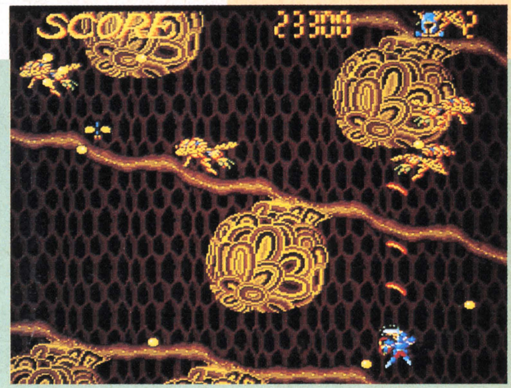
そのほか，敵の動きやアイデアなどにも，なかなか光るものがあった。このツールでは，決まりきった敵の動きや攻撃しか設定できず，状況で判断するような動きや攻撃ができないなかで，かなりのバリエーションを確保しているのは，センスと努力の組み合わせによるものだろう。なかでも姿を消して攻撃してくるボスは，ツールの機能をうまく使っていて，ハッとさせてくれるところが憎い。

また，最終面のあとにスタッフロールがあるのだが，これはわざわざ１面分のデータを使っていて，敵の代わりに文字を出すことで実現している。こういったツールを知りつくしたかのような演出を見ることで，この作品の完成度を確かめられるといっていいだろう。

個人的にいわせてもらえば，５面と６面はちょっとむずかしすぎるということぐらいが難点で，さすがグランプリというところではないだろうか。



顔，まさに中国の顔だ



空駆けるロードは虫と戦う

追い越せ負けるな，優秀作

残る２本の優秀作であるが，やはりグランプリよりは見劣りがするものになっているのは，しかたがないだろう。またグランプリが正統派だったので，逆にちょっと毛色の変わった意欲作が選ばれているという印象もある。

ひとつめは「三國志－幻伝－」という作品であるが，これはかなり異色の部類に入る作品で，タイトルと内容のギャップで思いっきり笑わせてくれる。写真を見るとわかるが，自機は玄徳の四角い顔で，敵もすべて四角い顔である。シミュレーションゲームに出てくるままの格好でシューティングをやらせるというセンスには驚嘆したが，全体的に見ると詰めが甘く，粗さが目につくのが，やや残念だ。

途中から舞台が宇宙になってしまうなどの荒唐無稽な展開ではなく，あくまで中華歴史ロマン路線を追求すれば，異様な存在感を与えたかもしれない。それほどこのセンスはすごい。しいていえば音楽をオリジナルにする（ぜひ中華風にしてほしい）とか，敵の攻撃の設定をもう少し詰めてほしいなどの注文はあるが，アイデアものとして異色の輝きを保っているのは事実だ。

もうひとつの作品は，「フェイバー・ザ・ロード」という作品である。こちらのほうは，騎士が空を駆け悪魔を倒すという，アリガチなRPGのような世界でシューティングをさせるものである。「ドラゴンスピリット」などと同じようなジャンルといえる。これはまさに無難な作りで，難易度も含めてそこそこ遊べるものになっている。中盤の，巨大な敵の体の上を背景として場面が展開していくところなどは，なかなかいいアイデアだ。可もなく不可もなくといったような印象が，こぢんまりとまとまっていて，好感を得たのだろうと思われる。

この作品も音楽は標準のものを使っているのだが，こちらはあまり雰囲気を損ねていない。しかしやはりオリジナルの音楽があれば，もっと印象はよくなると断言してもいいだろう。

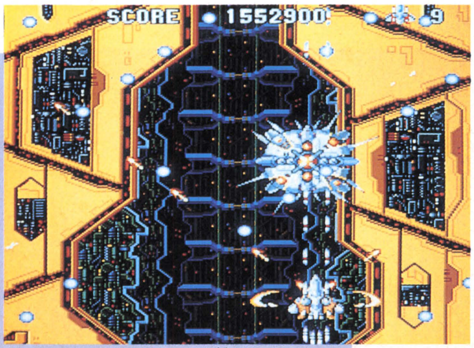
ツールの限界と作品

というわけで，各作品に固有の長所や短所を見てきたのだが，これらの作品すべてに共通の問題点がいくつか見受けられる。つまり，ツールとしての「シューティング68K」の限界のことだ。ショットの発射の優先順位のクセのため，連射をすると斜めや横の攻撃が出なくなったり，途中でディスクを入れ替える必要に迫られる（２ドライブなのに）ことである。

ユーザー側の工夫で克服できるようなものなら逃げ道はあるかもしれないが，こういったシステム上の問題点は，本体のバージョンアップによって対応してもらわなければ，どうしようもない。

しかも，今回のコンテストの作品によって，ツールとしての欠点や求められる機能がハッキリ示されたのではないだろうか。その貴重な情報をもとに，より親しみやすいツールとしての成長を希望したい。

誰もがゲームを作る喜びを味わえるために……。作ることの先には必ず何かがあるのだから。



消えるボス，かなりの強敵だ

負けずに作ってみるのがスジか？

とりあえず，これらの作品を見れば，「シューティング68K」がどれだけの機能をもち，どうやればどうなるかもわかるハズだ。ある意味，このツールで作品を作る人への教科書になるのかもしれない。ツール本体のエディタを使って，いろいろ解析してみることを勧める。持っていても宝の持ち腐れになっている人は，これを機に発奮してはいかがなものかと思う。

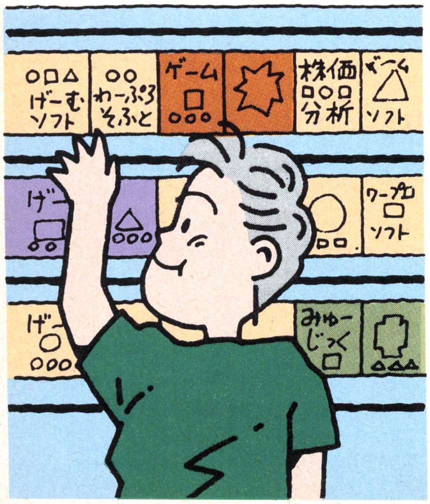
ちなみに，自分でこのツールを使ったらどんなゲームができるか考えてみて，ふっと浮かんだアイデアは，自機の弾を出さないようにしてしまい，上から見たカーレースにしてしまうというものだった（ログインのヨコスカウォーズを使ったゲームに似たものがあった気はするが）。ひまができたらやってみたいような気もする。シューティング以外にも応用性がきくとしたら，それはそれでこのテのツールの興味深い面白い可能性を示すことなのかもしれない。

総合評価

	0　5　10
ヴァリストレスナルト	★★★★★★★★
三國志－幻伝－	★★★★★★
フェイバー・ザ・ロード	★★★★★★
自分で作りたくなる	★★★★★★★

AFTER REVIEW

今回はレミングスです。派手なゲームではないのでパッと盛り上がったわけではないけれど，じわじわと人気が上がってきたゲームです。それだけに固定ファンも多いようです。



レミングス

▶なかなか解けないので長持ちするから。トラップにひっかかるのを見ているのがいい。　金井　英樹(21)神奈川県

▶はまる！（AMIGAとほとんど同じだったから）でもサンプリングが少なく音が薄い。　金村　厚男(18)広島県

▶レミングたちを集団自殺させると，気分はまさに「たまや〜」である。　加藤　洋介(25)埼玉県

▶手軽に遊べてよろしい。久しぶりに頭を使った気がするゲームだ。　西方　茂樹(23)茨城県

▶レミングたちがかわいいキャラで，動きもすばらしくきれいだったから！　あの小さいレミングたちが，ちょこまかと動いてリアルだ！　村上　政幸(18)愛知県

▶１面が短く手軽に楽しめる。しかし，パスワードを打ち込むのはちょっと……。　田下　昌充(20)神奈川県

▶一見，こまごましてわかりにくいゲームと思うが，一度やると奥の深さがすごく，キャラ１人ひとりに愛着がわいてくる。　辰己　真章(18)兵庫県

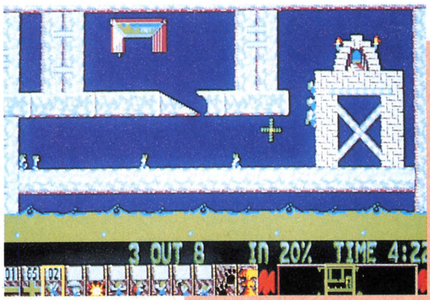
▶画面はシンプルなわりにのめり込んでしまう。やっぱりおもしろい。　林　寿弥(19)三重県

▶あっ，コラコラそっちに行くんじゃーない！　白井　崇文(22)神奈川県

▶Oh! No……ポンッ！　……やみつきになる。なかなか全面終わらないから。　藤井　実(21)千葉県

▶３人ぐらいで頭をひねくりまわしてやるのが，とってもナイス！　ロードランナーを思わせるようなレミングたちのグラフィック。　越智　亮(19)大阪府

▶面が進むとディスプレイ以外視界に入らなくなるほどのめり込める。　三輪　祐一(21)愛知県



▶シューティングのヘタな私には，このテのソフトしかない！　岩田　勝彦(19)岐阜県

▶レミングたちの愛らしい動きに愛着がわくからだよー。　高橋　昌弘(20)愛知県

▶なんといってもかわいいキャラクター，細かい動き。一度やると誰でもはまってしまうゲームだと思う！　和田　映二(21)滋賀県

▶シューティングもいいけど，頭も使わないと腐りますからぁ……。レミングたちがにくたらしいほどかわいい。　羽生　知浩(19)北海道

▶マニュアルのパスワードメモ帳をひとつずつ埋めていくという行動は，ラジオ体操に出てカードにハンコを押してもらう行為に通じるものがある。実に楽しくてしかたがない。　松本　拓司(18)埼玉県

▶いままでのゲームとは，どこかひと味違う新しさがあるゲームだ。　安田　稔(24)埼玉県

▶パワーモンガーより頭を使う！　タイムオーバーや集団自殺のときの悲痛な叫び声がたまらない（その声が聞きたくて思わず殺レミングをやってしまう……）。　市川　徳明(18)東京都

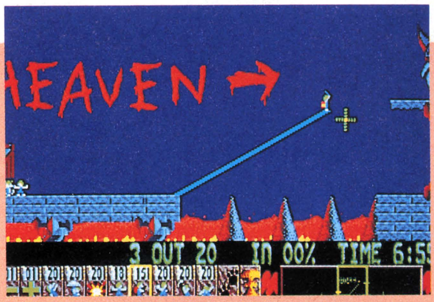
▶解けなかった面を解いたときの喜びとその快感を知ってほしいから。命を守る大切さを知ってほしいから（笑）。　小杉　貴秀(15)新潟県

▶スターウォーズといいレミングスといい，一度やってみると買いたくなりますね。　田高　浩一(25)東京都

▶ゲームは，このレミングスのように創造性があるほうがいい。シューティングゲームには破壊性しかないように思えてつまらないから。　香村　潤一郎(22)沖縄県

▶昔はまった「チャンピオンシップ・ロードランナー」を思い出させてくれた。　堀越　小千雄(21)東京都

▶あの“ムーミン”（に出てきたニョロニョ



ロ）を思い出させる。
根津　正博(26)福島県

▶レミングスやっと終わりました。Mayhemの29面がいちばんたいへんだったかな？　やり方はわかっているのに，タイミングがむずかしい面が多くて。最終面は“We all fall down”レミング100匹を期待していたのですが違いましたね。お年を召した方に頭の柔軟体操としていかがですか？
中内　英裕(28)栃木県

▶ポーズをすればじっくり考えることができるので，ロードランナーが苦手な私でもできたから。　片平　正二(17)福島県

▶１日中ヒマをつぶせてあきさせない。でも疲れてしまうけど。
北村　進二郎(30)滋賀県

▶アイデアがいいから。レミングたちの動きもユーモラスなこと！　パズルとしてもとてもおもしろい。　北内　啓(20)京都府

▶久しぶりに頭の体操ができるゲームだ。やり始めると必ずハマる。
後迫　浩一(31)東京都

▶文句なくおもしろいです。ただ，キャラクターが小さいので，ゲームをしたあと目が痛くなるのが難点ですね。
竹鶴　敏夫(18)広島県

▶土木関係者（土木工学科）からひと言。あの不自然な橋がよい。
小海　昌伸(18)新潟県

▶ゲームが苦手な人でもけっこう遊べる（これは自分のことだったりする）。
濱崎　健一(28)広島県

▶BLOCKERがかわいそすぎる〜。謙虚な姿勢のレミングたちがよい。
上池　宏幸(17)滋賀県

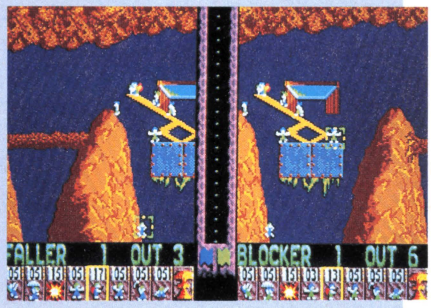
▶これは対戦がおもしろい。ポピュラスとはまた違った陰険な(!?)闘いが楽しめます。
金丸　勉(20)滋賀県

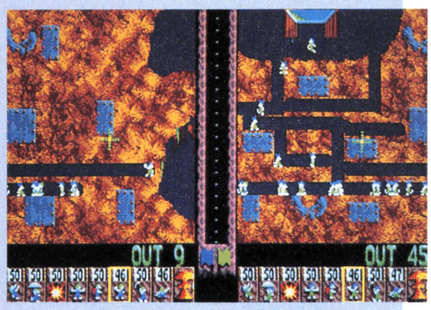
▶最近買ってハマってます。ちょっとしたヒマにできるから（でもハマるから結局……）。　小川　靖浩(20)東京都

▶８ドット，４色，８ドット，４色，８ドット，４色，８ドッ……。
岩浅　裕一(16)神奈川県

▶レミングスは悲壮なゲームである。私の友達は，大量に殺して喜んでいたが，あの声が夜中に私を苦しめる……。
小沼　健太郎(15)千葉県

▶このゲームは，パズルゲームにしてはものすごくおもしろいと思う。
羽深　修(16)秋田県





▶イマジニアのゲームはハズレがないと思っているから。　植木　正幸(23)神奈川県

▶とにかくムキになる。こんなにムキになったのはポピュラス以来だ。
海野　弘之(24)大阪府

▶いまわのきわのレミングたちが放つ花火が，このゲームを推すすべての理由です。
西崎　貴博(17)北海道

▶はっきりいって，ポピュラスの比ではない！　対戦にハマりまくりました。
宮原　大(19)岡山県

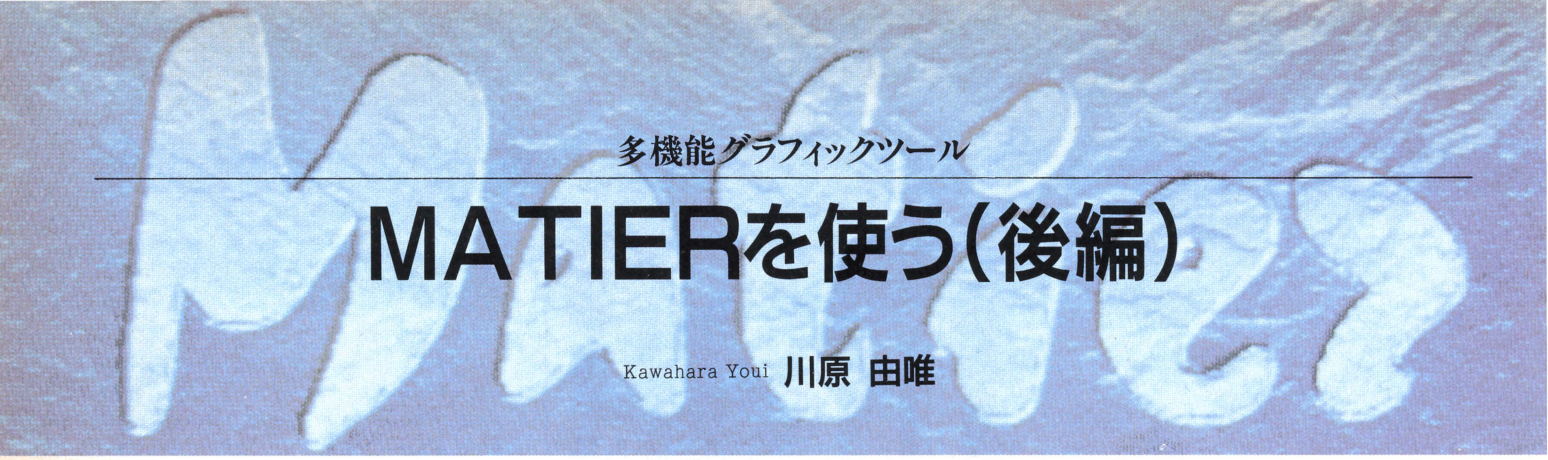
▶キャラクターの動きがたいへんユニークで，かつよくできているゲームだと思うから。　浅見　喜代司(21)群馬県

発売中のソフト

★ファイナルファイト　カプコン
X68000用　5″2HD版　9,800円(税別)
★リーディングカンパニー　光栄
X68000用　3.5/5″2HD版　12,800円(税別)
★ヴェルスナーグ戦乱　ファミリーソフト
X68000用　3.5/5″2HD版　9,800円(税別)
★シューティング68K GAMES（ヴァリストレスナルト）
ブラザー工業(TAKERU)
X68000用　3.5/5″2HD版　3,000円(税込)
★シューティング68K GAMES（三國志一幻伝一，フェイバー・ザ・ロード）　ブラザー工業(TAKERU)
X68000用　3.5/5″2HD版　3,000円(税込)

新作情報

★ポピュラスII　イマジニア
X68000用　5″2HD版　価格未定
★エトワールプリンセス　エグザクト
X68000用　5″2HD版　価格未定
★ドラゴンスレイヤー英雄伝説　SPS
X68000用　5″2HD版　価格未定
★デスブレイド　SPS
X68000用　5″2HD版　価格未定
★ファルディア　M.N.Mソフトウェア
X68000用　5″2HD版　価格未定
★究極タイガー　金子製作所
X68000用　5″2HD版　価格未定
★エアバスター　金子製作所
X68000用　5″2HD版　価格未定
★バーンウェルト　グローディア
X68000用　5″2HD版　価格未定
★エアーマネジメント　光栄
X68000用　5″2HD版　11,800円(税別)
★ライフ・イズ・ミュージック　光栄
X68000用　3.5/5″2HD版　価格未定
★沈黙の艦隊　ジー・エー・エム
X68000用　3.5/5″2HD版　12,800円(税別)
★ネクタリス　システムソフト
X68000用　5″2HD版　予価7,800円(税別)
★OVERTAKE（仮）　ズーム
X68000用　5″2HD版　価格未定
★ふしぎの海のナディア　ガイナックス
X68000用　3.5/5″2HD版　14,800円(税別)
★キャッスルズ　ビクター音楽産業
X68000用　5″2HD版　9,800円(税別)
★ライジングサン　ビクター音楽産業
X68000用　5″2HD版　9,800円(税別)
★ウェルトリス　BPS
X68000用　5″2HD版　7,800円(税別)
★サークII　ブラザー工業(TAKERU)
X68000用　3.5/5″2HD版　8,800円(税別)
★チェイスH.Q.　ブラザー工業(TAKERU)
X68000用　3.5/5″2HD版　7,800円(税込)
★餓狼伝説　ホームデータ
X68000用　5″2HD版　予価8,500円(税別)
★サバッシュII　ヒッパロスの風
ポプコムソフト/グローディア
X68000用　5″2HD版　12,800円(税別)



多機能グラフィックツール

MATIERを使う（後編）

Kawahara Youi 川原 由唯

MATIERは質感重視の2D処理，高機能エディット＆3D処理という“HIGH TEC＆HIGH TOUCH”を実現した新しいグラフィックツールです。

TUBEの季節！　されど蒸し熱いなかに中嶋美智代の新譜を流してさらに頭をクラクラさせながらこの原稿を書いている今日はまだ梅雨明け前であったりする（7月19日現在・東京）。あぁ，夏だね。

＊　＊　＊

久しぶりに手放しで誉めまくれるツールに出合えた。その名をマチエール（MATIER）という。たぶん，材質や質感といった意味の，（一般には）芸術用語。英語でいうところのmaterialと同義と思う。辞書に載ってないから自信はないが，たぶんフランス語だろう（Matiéreの男性形？）。

先月号の紹介記事・前編では，中野氏が全体的な機能と環境について紹介されていたので，今回は実践編ということで，ちょっとしたサンプルなどを載せながら，具体的な特徴を紹介してみよう。

CGっていったい

さて，皆さんが“CG”と聞いてまず思いつくモノって，どんな感じのものでしょう。カーグラフィクスなんて10年前のギャグいってる奴はその辺にうっちゃっといて，大半は「アニメ（セル画）調の絵」や「レイトレーシング」のような3D CGを思い浮かべるんじゃないかな。セルアニメはオタク文化の代表格だから，我々のような最先端ヲタクに絵を描かせれば，それがラムちゃんにケイトーしてしまうのはもっともなことだし，3D CGはコンピュータがなかったらまず絶対に生まれなかったんだから，思いつくのは当然だね。

「CGやってます」と自己紹介なんかしたら「アカデミックな方なんですねぇ」なーんて評判になってしまうようなところも，一般にはあったりする（実はそれが乱馬の絵でも）。

いずれにしても，新しい文化，デジタルな触感を持ったアートの代表として「CG」という言葉は存在するように思う。

最近になって，人並みの市民権をそれなりに確立してきてからは，CDジャケットや，広告関連のアイキャッチングイラストなど，ちょっと先行きたいおサレ業界のステイタス的利用が増えてきてはいるが，でもそこいら辺での利用って，たいていは取り込み写真の加工品であるとか（レタッチなどともいう。これがたいていMacintosh使ってるんだなぁ），ただ“コンピュータを使った”ということが意味を持つコンピュータ自身の広告などでしかなかったように感じる。純粋なCGってものの定義がどんなもんかは知らんが，取り込み画の加工品で

さまざまなペンで作られたブラシによる表現の違いを見てほしい。それぞれ右側が基本となるブラシパターン，左側の犬が応用例だ。順に，
オーソドックスな透明水彩風
ちょっと透明度を下げた不透明水彩風
細いペンでストロークを強調した色鉛筆風
透明度を上げた淡い水墨画風
ランダムなベタ塗りストロークの油絵風
といった感じ。ブラシの設定は簡単でもっと多彩な表現も可能だ

はイマイチCGとしては認めたくないな……ってな気がする。

しかも安っぽい作品が多いゆえ，コンピュータグラフィックの「アート」としての認知度は一般には低いようだ。個展を開いたりする作家も徐々に現れ出したが，絶対数からしてまだまだ少ない。実際，評価に値する作品はほとんどない，特に二次元作品においては（と，いいきってしまってもあんまり罪悪感を感じないのが困ってしまう）。

さて，CG，CGと簡単にいっているが，CGっていったいどんなモノだろう。どーいうわけか，たいていのCGって，ひと目でCGだということがわかってしまうよね。

なぜだろう。いままでのCGの特徴として，輪郭線がはっきりしている。ベタ塗りが多い。原色が多く使われる傾向がある。つやつやしてる。てかてかしてる。ぬるぬるしてる……云々。

いささか抽象的な言葉が多いけど，実際にCGの特徴を抽出すれば，上記のような表現でいい表すのが簡単だよね。これが好きだという人もいれば，「硬」い，「冷」たい，からキライだという人もいる。

対して普通の絵画の特徴ってなんだろう？　画法によるところも多いけれど，画家の個性を表すいちばん重要なファクターはやはり「タッチ」だといえないかな（逆にいうと，大量生産流れ作業型テレビアニメがなぜセルアニメを採用しているかといえば，セル画が個性の出やすい「タッチ」を最小限に抑えられる画法だからだろう）。

そう，CGと一般の絵を分け隔てているもっとも重要な点が「タッチ」なのだ。油絵や水彩の温かみというのは，筆やペンのタッチによるところが大きい。

自然画対応

筆やペンのタッチを表現することのできるCGツールは，現在では，まだ少ない。ベタ塗りが綺麗にできるCGだったからこそ，特にパソコンではアニメ調の絵が流行ったんだと思う。ツールの限界が大きな要因であったはずだ。決して両マニアの相関が高かったという理由だけではない（それもあるけど）。

また，話はちょっと飛躍するけれど，い

ブラシを使った草原の表現。しかし，ファイルサイズは400Kバイトを超えた……

ままでのツールが，既存の画材の束縛から抜け切れていないということもいえると思う。「水彩風」「油彩風」であるとか，「パステル風」とか，すでにある画法の雰囲気を描くことができるツールがよいもののような評価をされがちだ。でも決してそういうツールがいいツールなんじゃないよね。CGだからこそできる表現を持ったツールが，実は待ち望まれている昨今なわけだ。

そういった意味では，この新製品「MATIER」は，両方の要求を満たしてくれるかなり優れたヤツだ。色変換やトランスフォームといったCG特有の機能はもちろん，いままでのCGツールでは表現することが難しかった「タッチ」を簡単に出すことができるのだから。僕が知る限りでは，X68000の6万色モード使用のツールでこういった微妙な筆使いをシミュレートできるものはいままでなかった。2D CGの新たな時代の幕開けだ！

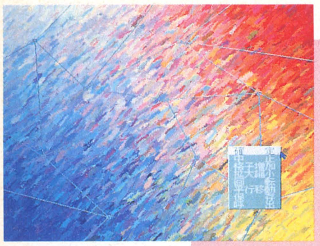
いろんなペンがついておトク

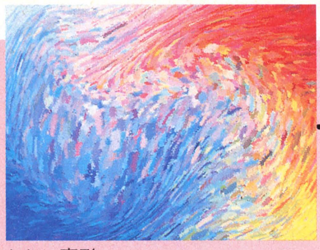
もうついていて当たり前の機能は紹介しても無駄だからしない。おいしくて強くなるなる機能だけピックアップするぞ。

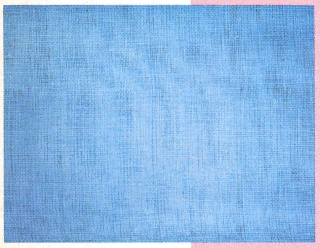
まず紹介したいのがさまざまなペン先だ。とにかくMATIERはペン先のバリエーションが豊富，というか，組み合わせることで無限のパターンを生み出すことができる。この組み合わせがあまりに多くて覚えきれないくらいだ。特に魅力的なのがブラシ。ブラシといっても，ただのエアブラシではない。ブラされるのが選択されているペンなのだ。そのペンってのがうねうねだったりくにょくにょだったりふわふわだったりぽちぽちだったり（説明になっとらん），あぁ，言葉では表現しにくいのだ。百読は一見にしかず。画面写真を見ておくれ。これだけでMATIERの凄さがわかると思う。

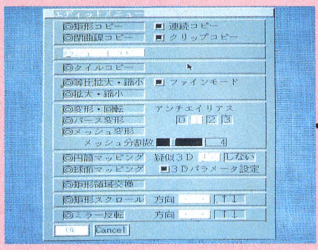
毛筆モードは難しい。フラクタルペンはもっと難しい

メッシュ変形の効果

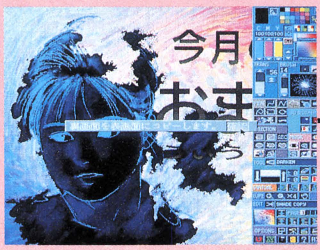

操作ひとつでこんな模様も

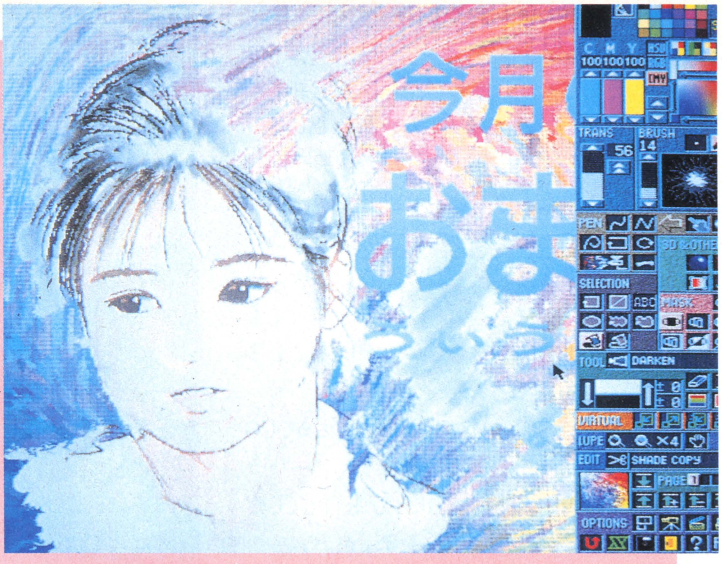

さらに変形


布地を取り込み


シェイドコピー

基本の絵に……


マスクをかける


さまざまな処理を組み合わせて画像を合成してみた。まず，グラデーションスプラッターブラシという特殊効果で，画面全体に4点グラデーションの色でブラシをかける。ストロークの長い油絵調の元絵はメッシュ変形で簡単にゴッホ，ムンク風にできる。次にスキャナから取り込んだ布地でシェイドコピーを行う。バンプマッピング風の質感が出る。そして，大本となる絵の背景をマスク指定し，裏画面との合成を行う。透明度は56%で合成されている。MATIERは豊富な裏画面（ただし4Mバイト以上）と賢いマスク機能による画像合成や豊富なエフェクタによるフォトレタッチ処理に力を発揮する

正直なところ，これを使いこなすのはちょっと大変だなって気がするが，機能の使い方自体に難しいところはないし（エディタ一発），これだけの重そうな処理を，不思議なくらい気持ちのいい速度でこなしている。十分実用レベルだ。うれしい。

フラクタルペン（などといっているが，要するに「にじみペン」である）もある。中心部が濃いめ，周辺部が薄めに，ランダムな広がりをするペンだ。分岐数などパラメータ調整で太さ（?）を調節することもできる。フラクタルというだけあって，粗い紙の繊維に絵の具がにじんでいくような跡が得られる。水彩っぽい表現が可能だ。

水彩といえば，忘れてはならない機能がある。それが透明ペン（?）だ。これはすでに画面上にある色の上を，水をつけた筆でこするような効果をだせる。また，透明色だけではなくて，色の着いた絵の具でこすることも可能だし，パステル画のように紙の上に載っている絵の具をこすって周りに広げるような効果もできる。にじみペンの後処理版と考えてもいい。これも実にオイシイ機能。ほしかったんだ，こういうの。

特殊機能

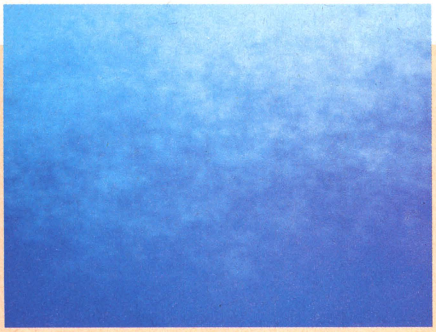
特殊機能にもなかなか面白くて使える機能がつまっている。特にメッシュトランスフォームは圧巻。いままでのツールでもトランスフォーム機能を備えていたものはあったが，せいぜい台形変形くらいにしか使えなかったよね。ところがMATIERのそれは，矩形領域の要所に制御点を設定して，その制御点の移動によって，ちゃあんと間を補間して滑らかに変形してくれるのだ！　う〜む，楽しい。これぞCGの醍醐味ってやつよね。これからのお見合い写真には必需品かもしれない機能（うそ）。

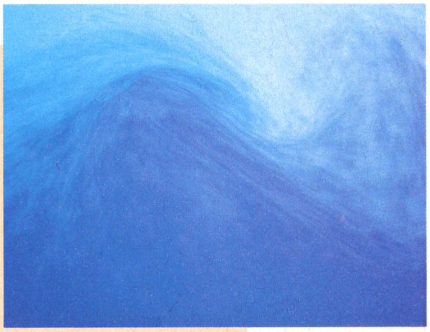
あと，ランダムノイズを入れる処理。これも面白い。画像の圧縮率は悪くなるが，アニメ調のベタ絵に入れるとざらっとした自然画調の雰囲気になる。使える。

平滑化・鮮鋭化・輪郭抽出・レリーフ・モザイク・フレアなどの，Z's-EXでお馴染みの機能もデフォルトで搭載。

拡大・縮小・ネガ・ポジ・アンフォーカスは，もはやついていて当たり前，だがそれぞれがオーバーサンプリングによるデジャギーを行っているので，非常に高品位な結果が得られる。グラデーションはディザ設定可能でマッハバンドのない美しい仕上がり。

あと，前回に紹介されていた球体をフォンシェーディングで描く機能や，もっこし機能。これを見てたら，昔流行ったTシャツなんかに塗ってドライヤーで熱風を当てると「もこもこもこ」と膨らむアレを思い出してしまった。ま，使い方によっては面白いやね。球体フォンシェーディングは，あらかじめ裏画面に転送しておいた絵を球


これはZ's-EXで描いた雲……


メッシュ変形でこんなに表情が出る

状にマッピングすることもできる。水晶玉みたいだ。

弘法だって筆選び

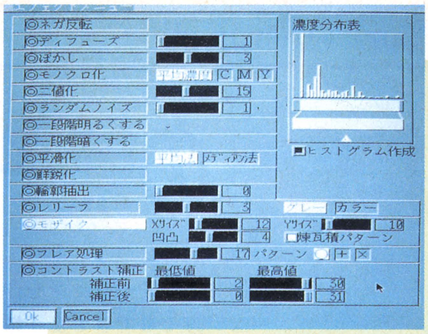
いまどきのCG絵描きは，2，3種類のツールを使い分けるのが当たり前になってきている。ひとつのソフトで自分に必要な機能をすべて備えているものなんて（自作でもしない限り）ありえないし，限られた環境下では，それぞれのソフトの得意なところを組み合わせて使うのが効率的だからだ。僕自身，これからはMATIERとZ'sの両方を使っていくつもりだ。はっきりいってこいつには惚れた。

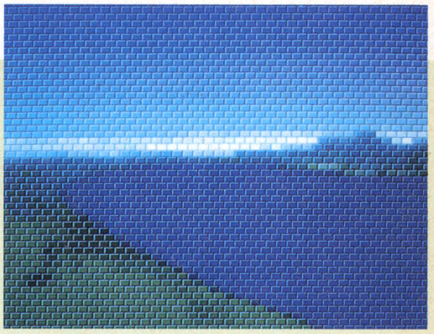
MATIERは，とにかくアーティスト寄りの設計がなされたツールだ。したがってアニメ調の絵を描く絵描きがMATIERを選ぶのはあまり意味がないことかもしれない（ツールに流されてみたい，使われてみたいってなら，ちょっかい出してみるのも経験だけど）。タイリングペイントがない分，アニメ絵は不得手。微妙な筆遣い，一筆入魂の絵師が使って初めて意味を成すツール……，おおげさかもしれないけど，本当にそんなふうにいってしまいたいところがある。「俺のマウスさばきはニッポンいちや」いうちょっと自信過剰気味の人なら，悪いことはいわない。買いなさい。これ使わなきゃ損だよ，絶対。いままでZ'sを使っていて不満があった人もぜひ一度お試しを。ひょっとしたらあなたのよきパートナーとなってくれるかも。

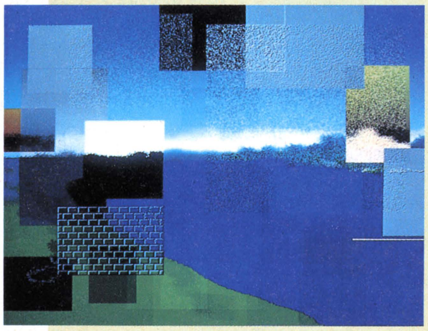
「買い」なんだけど

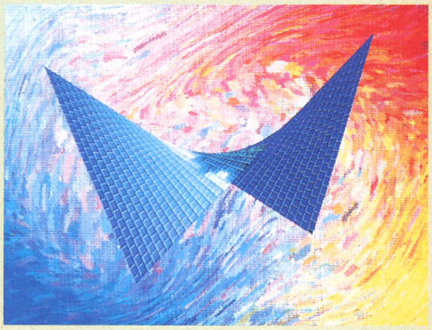
「凄い，凄すぎる」。あまりの多機能ゆえに，紹介できなかった機能も多く存在する。しかも筆者が普段，ガキっぽい絵しか描いていないもので，力及ばず，チャチなサンプル画しか描くことができなかった。今回はツールを前にして「石になる」という気分を本気で実感してしまった。使いこなせるようになるにはまだまだ修業が必要のようだ。人生是修業よのぉ。

とにかく，MATIERの特筆すべき点は「タッチ」である。ほかにも面白い特殊効果が山盛りなのだが，あまりにもタッチが凄いのでそればかり強調してしまった。







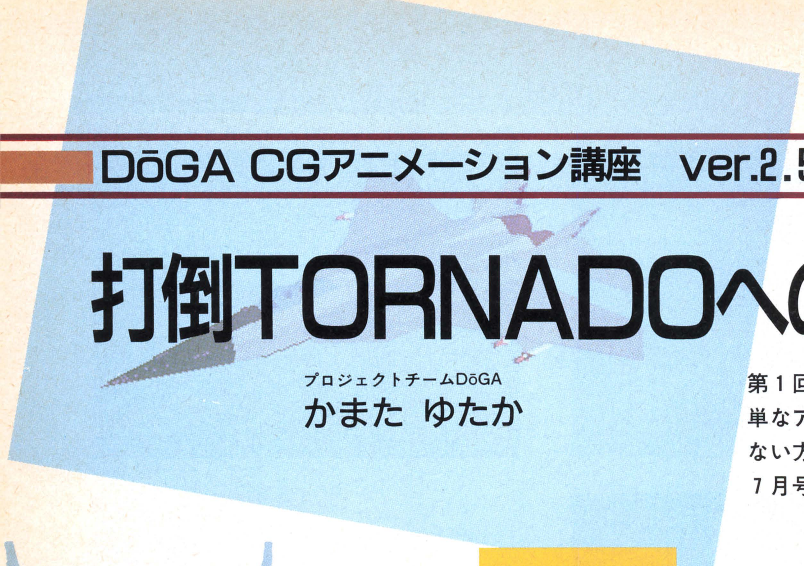


エフェクト＆エディット機能の競演。モザイク処理には「タイル化」というオプションがある。なにかと思えば……，文字どおりタイル壁画に変換してしまうのだ。自由変形などを見ても，変形画像は美しい。オーバーサンプリングの効果が十分に表れている

それにしても，豊富なタッチと特殊効果のおかげさんで仕上がった絵の圧縮率の悪いことひどいこと。この調子だとPICフォーマットで400Kバイト超えるのなんて当たり前の時代がくるかもね。

真面目にMATIERを使いこなせる絵師が現れ出したそのときは，プロアマ問わずX68000CG界もなかなか侮れない存在になることだろう。やがて，MATIERがX68000のグラフィックツールの代表としての地位を確立（絶対に確立するだろうけど）したら，画像フォーマットの標準が現在のPICから，自然画に強いJPEGフォーマットのようなものに変わっているかも。5年目にして，ますます面白くなってきたX68000CGだ。わくわく。

MATIER　39,800円（税別）
サンワード　☎044(855)4335



打倒TORNADOへの第一歩(前編)

プロジェクトチームDōGA
かまた ゆたか

第1回目ということで，お試しシステムのデータを利用して，非常に簡単なアニメーションを作ってみましょう。マニュアルをまだ入手していない方でもできるように，すべての操作を具体的に記載しました。さあ，7月号の付録ディスクを取り出して，あなたもCGAアーチスト！

はじめに

今回から始まるこの連載は，本年の7月号の付録ディスクとして配布した「DōGA CGAシステム」を用いて，皆さんがCGアニメーション作品を制作できるようになることを目的としています。DōGA CGAシステムや当チームの解説は7月号をご覧ください。また，7月号を持っていない方のために，CGAシステムの入手方法については次ページのコラムにまとめてあります。

さて，マニュアルの申し込み用紙の裏の自由記入欄を拝見すると，連載は，初心者でもわかるような初歩の初歩からやってほしいという声が多いようですので，当分の間はマニュアルの「CGA大学編/教養課程」並の内容にしたいと思います。

制作の手順は，可能なかぎり具体的に解説しますので，とりあえず書いてあるとおりに実行してみてください。そして，そのあといろいろ試行錯誤することとで，テクニックを磨いてください。パワーユーザーには少しものたりないかもしれませんが，楽をするためのテクニックや面白い表現方法なども盛り込んでいきますのでお見逃しなく。

また，2カ月遅れのペースで“補習”のコーナーも設けようと思いますので，わかりにくかった点や疑問などがありましたら，DōGAプロジェクトルーム内「CGA講座補習係」までお手紙ください。

今回の目標

芸術祭のグランプリ作品「TORNADO」はご覧になりましたか？　7月号では，写真がたくさん掲載されましたが，やはり静止画では，実際の作品のスピード感，迫力，感動は伝わりにくいものです。

ところで，制作者である文月涼さんは授賞式で，“私が特別なのではなく，X68000とCGAシステムがあれば，誰にでもこういった作品ができるってことを知ってもらいたい”と語っていましたが，さて，これは本当でしょうか？　誰にでも，つまり，まったくの初心者でも「TORNADO」はできるのでしょうか？

“なにむちゃをいってるんだ。CGの知識も経験もないド初心者の自分にできるわけないだろ。文月さんは単に謙遜しているだけだ”と思う方も多いでしょう。本当に，ただの社交辞令？

この問題の結論は先送りにして，実際に皆さんに数カットのCGAを制作していただきましょう。まず，今回の前編でFFEとAUTOだけを使って，視点だけが動く非常に簡単なアニメーションを制作します。画質を向上させる方法も習得しましょう。

題材は「TORNADO」に対抗して，「お試しシステム」のサンプルデータのF1を使います。

そして次回の後編では，複数の物体が動くカットを制作して，最後に「お試しシステム」のデモ画像と一緒に，連続アニメーションとして作品風にまとめます。

連載1回目から，いきなり「TORNADO」とタメを張ろうというこの企画はちょっと無謀ですかね。でも書いてあるとおりに実行すれば，必ず誰でもできるはずです。

とりあえず1カット制作してみる

以下の解説は，フロッピーディスクでCGAシステムを使用していると想定しています。ハードディスクで使用している場合はドライブ名などが異なってきますので，ご注意ください。

○準備

概要：制作に入る前に，必要なものを揃えます。

操作：1)　7月号付録ディスクから解凍した，
　　お試しシステム
　　CGAシステム
　を用意する
　2)　データディスク用にフォーマットずみのブランクディスクを用意する

解説：このレクチャーには，およそ30分の時間を要しますので，そのくらいの時間も用意してください。フォーマットの方法もわからないという初心者の方は，CGAシステムのマニュアルの「CGA大学編/教養課程/パソコン基礎概論」の単位を習得してから，再チャレンジしてください。

○PESを終了する

概要：CGAシステムのウィンドウシステム型のメニューであるPESを終了し，コマンドラインから操作できるようにします。

操作：1)　ドライブ0に「CGAシステム」を入れ，ドライブ1にデータディスクを入れて，「OPT.1」を押したまま電源を入れる

→CGAシステムが起動して，PESの初期画面になる（図１）

2)　マウスの右ボタンを押す

→ポップアップメニューが出る（図２）

3)　右ボタンを押したまま，ポップアップメニューのいちばん下の「終了」にマウスカーソルを持っていって，ボタンを離す（図３）

→CGAシステムが終了し，「B>」と表示される

解説：安心してください。PESを終了しても，CGAシステムを使うことはできます。PESは，各コマンドの実行などがマウスひとつで楽々操作できるため，初心者にとって便利なツールですが，PESの使い方を一から解説していると，それだけで連載が１回終わってしまいます。ですから，今回はPESを使わずにコマンドラインからCGAシステムを実行します。PESの使い方は「CGA大学編/教養課程/PES基礎概論」をご覧ください。

○データディスクの作成

概要：「お試しシステム」から必要なデータファイルをコピーします。

操作：1)　ドライブ0の「CGAシステム」を「お試しシステム」と取り替える

2)　「B>」のあとに，キーボードから，

copy a:¥test2¥object¥tyrl.＊

と入力し，リターンキーを押す

→以下のように表示される

```
a:¥test2¥object¥TYRL        .ATR
a:¥test2¥object¥TYRL        .DOC
a:¥test2¥object¥TYRL        .PIC
a:¥test2¥object¥TYRL        .SUF
 4個のファイルをコピーしました
```

3)　「B>」のあとに，キーボードから，

copy a:¥test2¥background¥jimen.＊

と入力し，リターンキーを押す

→以下のように表示される

```
a:¥test2¥background¥JIMEN   .ATR
a:¥test2¥background¥JIMEN   .SUF
 2個のファイルをコピーしました
```

解説：「TYRL」というのは，「お試しシステム」に入っていたF1のデータで，「JIMEN」は背景用のデータです。ここでは６つのファイルをコピーしていますが，今回は「TYRL.DOC」と「TYRL.PIC」は使用しません。

○FFEを起動する

概要：モーションデザインツールFFEを起動します。

操作：1)　ドライブ0の「お試しシステム」を「CGAシステム」に戻す

2)　「B>」のあとに，

FFE

と入力し，リターンキーを押す

→FFEが起動し，初期画面が現れる（図４）

○物体設定

概要：「TYRL」を呼び出し，移動せずにそのまま原点に置きます。

操作：1)　メッセージパネルの「4)　物体設定」を左クリックする

→物体設定のメニューが出る

1)　追加

2)　変更

3)　削除

4)　終了

2)　「1)　追加」を左クリックする

→形状ファイルの入力モードになり，２つの形状データが表示される

TYRL.SUF

JIMEN.SUF

3)　「TYRL.SUF」を左クリックする

→「データ読み込み中」「処理実行」などが表示される

図１　初期画面

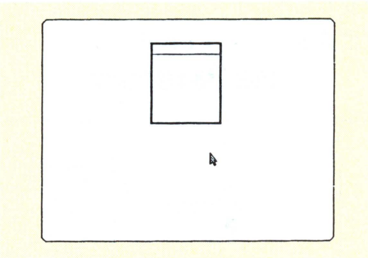


図２　ポップアップメニュー

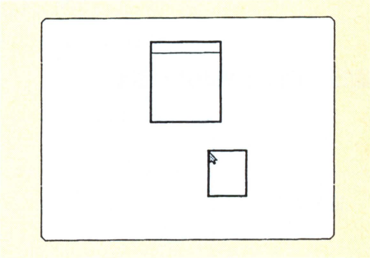


図３

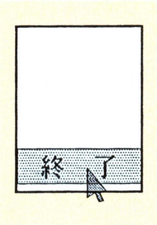
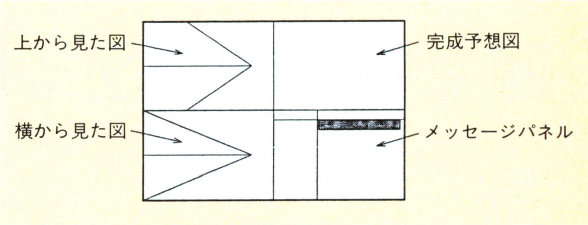


図４　FFE初期画面



CGAシステムの入手方法

そんな人がたくさんいるとは思いませんが，７月号をたまたま買い忘れたとか，今月号からOh!Xを読み始めたという方のために，CGAシステムを再配布します。

７月号を持っていないということは，当チームのことも，CGAシステムのことも知らないでしょう。簡単に解説すると，当チームは，プロのソフトハウスではなく，阪大，京大のコンピュータクラブを中心にした，CGアニメーションの共同開発チームです。そして，そこで開発された一連のCGアニメーション制作プログラムが，CGAシステムです。

ということで，別に売って儲けようという意図はなく，当チームの主旨に賛同するアマチュアの方にかぎって，実費（大部分がマニュアル代）＋カンパで配布しているわけです。配布は，非常に手間がかかって，当チームの負担になっていますので，あまりむちゃをいわないようにご協力お願いいたします。

○申し込み期間

1992年８月１日～９月30日

○内容

CGAシステムディスク　　１枚

マニュアル（800ページ）　１冊

○費用

実費（2,000円）＋カンパ（１口1,000円，１口以上）

○申し込み方法

住所，氏名（フリガナも），電話番号を明記のうえ，「CGAシステム（ディスク＋マニュアル）がほしい」と書いて，プロジェクトルームに送る。後ほど，こちらから振り込み用紙などを送ります。

○申し込み先

〒533　大阪市東淀川区淡路５－17－２　102号

プロジェクトチームDōGA「めんどうくさいCGAシステム配布係」

その後，F1が黄色で表示される（図５）
4) 「作画」を左クリックする
→右上に完成予想図が表示される
5) 「決定」を左クリックする
→左側の２つの図面のF1が水色に変わる（図６）
メッセージパネルが物体設定のメニューに戻る
6) 「4) 終了」を左クリックする
→メインメニューに戻る

解説：物体を追加すると，まず原点(0,0,0)に表示されます。通常は追加すると同時に位置も設定しますが，今回はこのまま原点に置いています。位置の変更や複数の物体を設置する方法は次回で解説します。

○フレームNo.の変更

概要：最初はこの位置，何秒後はこの位置……というふうに，時間と位置を交互に指定して動きを設定します。時間はフレーム単位で設定します（１秒は20フレーム）ので，まずフレームNo.を設定します。

操作：1) メッセージパネルの「5) フレームNo.設定」を左クリックする
→フレームNo.設定状態になる
ナンバー： 1
決　定
削　除
中　止
ナンバーの「１」が反転して，入力可能状態であることを示している
2) キーボードから「20」を入力し，リターンを押す
→「このフレームはまだ設定しておりません。新しく設定することができます」と表示され，「決定」が反転する
3) リターンキーを押す

図5 TYRL読み込み直後

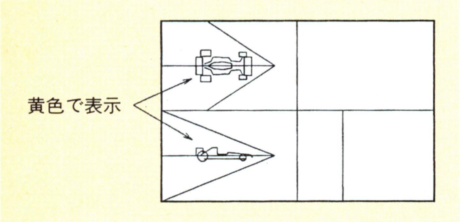


図6 物体設定決定

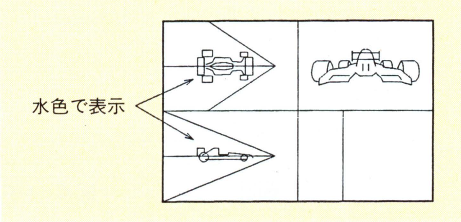
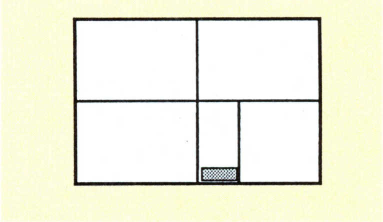


図7 フレーム数



柚姫のあちゃちゃちゃCGA

こーんにちは，柚姫です。今日は大阪市内の大きなお寺でお茶の会があって，そのお手伝いに行ってきました。300人近くのお客さんが入れ代わり立ち代わりにやってこられるので，もうたーいへん。姫も最後のほうにお点前をしたのですが，めっちゃくちゃ緊張しちゃいました。こんなに大きな会は初めて。手が震えて，抹茶の粉がうまくすくえなくって……。

だいたいお茶席っていうと，長い正座がつきものなんですが，これには暗い過去があるんですよね。

ずっと前の会のときのこと，お点前がすんだのでしまおうと立ったつもりが，立てない……，なんか足元が変。あれれ，なんかおかしい。ふと見ると左の足の甲が下を向いている。と，思ったとたんに後ろ向きに転んで……，ひぇ～。

姫は立ち上がって，そのまま何事もなかったかのようにしずしずと部屋を出ました。だけど，お客さんは必死で笑いを噛み殺しているし。は，恥ずかしかったよ～。

今回は「専門課程/第１限 ATR基礎実習」だ～

このコラムは，マニュアルの「CGA大学編」に沿って勉強する予定でしたが，こつこつやるのが苦手な姫は，いきなり飛び級してしまいました。だけどいいのかな（ちょっと不安な姫）。

えーっと，PESを立ち上げて，あっ，フロッピーをフォーマットしとくの忘れた。PESの中でフォーマットできないかなぁ，うーん。うろうろ，うろうろ。で，できない，ぐすぐす（Human68kに戻る姫。でも，最近はもうフォーマットが躊躇しないでできるようになったので，ちょっとうれしい）。

で，またPESに戻って。それからコマンドウィンドウでATRを選択する（適当にマニュアルを斜め読みしながら始める無謀な姫）。えーと，形状ファイルウィンドウから「JETKI.SUF」を指定すればいいのね。あれ？　物体がなーい。

そりゃ，フォーマットしたディスクには，なーんにもないのは当たり前？　そうかなー。そういわれればそんな気もする。でも，どうやったら物体が出てくるのかな（ちゃんと読まないですぐに人に頼る姫。しかし世間の風は冷たい……）。

そっか，お試しディスクの中から，ジェット機と地面の形状データをコピーしておかないとだめなんだ。ぐすぐす（またHuman68kに戻る姫）。そうだ。コピーしてる間に，明日使う花火を買いにいってきまーす。明日，お客さん（女の子）がたくさん来るっていっていたから，いまから楽しみ。いっしょに花火ができたらいいな（ATRを起動する前に，すでに現実逃避している姫，原稿は間に合うのかな？）。

マッチャマチにて

ほかのスタッフの人たちと一緒に，松屋町まで花火を買い出しにいく。えーっ，こんなにたくさん買えるんですかー。や，安い！　さすが浪速の商人（あきんど）は違う。ほくほく（妙に感動している姫，でも姫は生まれたときから大阪人）。原稿はあがってないけど，すでに幸せ～！

いとしこいしのJETKI.SUF

やーっと，戻ってきました。それで何をするんだっけ。すでに，心は花火とともに遥か彼方に飛び去っている姫。あ～した天気にな～れ。そうだ，そうだ。「JETKI.SUF」を読み込みながら，ATRを起動するんだ。

なんでー。どうして形状ファイルウィンドウに何にも出てこないのよ。ちゃんとコピーしたのに。わかんないよ。花火に心を移した私が悪かった。あれはほんの出来心。帰ってきてくれー，「JETKI.SUF」やーい（気分はすっかり妻に逃げられた夫のよう）。

またまたHuman68kを行ったり来たり。だけど，物体は出てこない。う～ん，どうやらなんか違うみたい。明日にしよーかなっと。全然解決になってないなあ。

（そこへ神の声が……）なんだ，なんだ？　PESを起動したドライブが違うって？　カレントドライブを変更したらいいのか（神の声：ちゃんとマニュアルを読め～）。あっ，「JETKI.SUF」だ。やったー。感激，うるうる。

ATRを起動できたら，あとは色なんかを変えていくだけ。前にもやったことあるから，簡単，簡単。とりあえずマニュアルどおりに変えてみよっかな。

ということで，今月は，アトリビュートのデータを変更するとこで終わ～り（それは最後の数行だけという気もする）。

今月の諸注意

姫のようにならないように，
1) マニュアルはちゃんと読みましょう
2) PESを使用するときは，カレントドライブに注意しましょう
3) 長時間の正座には気をつけよう
じゃ，またね～。

→メインメニューに戻る

現在のフレーム数の表示が,「20」となっている(図7)

解説：フレームNo.が20，つまり１秒後の設定を行おうとしているわけです。現在のフレームNo.は，常に図７の位置に表示されています。ところで，FFEの作者である三保君は仕様だというのですが，なぜかこの「フレームNo.設定」ではマウスが使えません。「決定」を左クリックしても無視されます。「削除」や「中止」の場合，カーソルキーで選択して，リターンキーで実行してください。

○視点の変更

概要：

フレームNo.が20のときの視点を設定。(1000,0,0)にあった視点を(-400,800,600)に変更します。物体の位置は，今回は変更しません。

操作：1) 「3) 視点設定」を左クリックする

→視点設定状態になる

	視点	注目点
X座標	1000	
Y座標		
Z座標		
画面回転		度
画 角		度

作　画
決　定
中　止

(X座標の「1000」が反転表示されている)

2) キーボードから「-400」と入力し，リターンキーを押す

→X座標が「-400」となる

Y座標の入力状態になる

3) キーボードから「800」と入力し，リターンキーを押す

→Y座標が「800」となり，Z座標の入力状態になる

4) キーボードから「600」と入力し，リターンキーを押す

→Z座標が「600」となる

5) 「決定」を左クリックする

→完成予想図が変化する（図８）

メインメニューに戻る

解説：各図面でのX，Y，Z軸の向きは，図９のようになっています。視点，視線は常時，赤と紫の直線で表示されています（図10)。

今回は視点を動かしましたが，同時に注目点を変更することももちろんできます。

読者によるほっとけないほっとこらむ

このコーナーは「柚姫の明るい悩み相談室」のあとを継いで，読者の皆さんから寄せられたご意見，ご感想，質問など，ホットな生の声を紹介します。

はじめまして，うさ子と申します。現在，某S社でOLをしています。数年前に阪大コンピュータクラブに在籍していたというだけの理由で，なぜかこのコーナーを担当することとなりました。だけど正直いって，まともにCGに取り組んだこともなく，横からちゃちゃを入れている程度だったりして。未熟者ですが，よろしくお願いいたします。

さて，今回はマニュアル申し込み用紙の自由記入欄から紹介いたします。

<お試しシステムの感想>

「素晴らしいアニメにびっくりしています」

「このようなデータを作成する人は，“神”か“超ヒマ人”のどちらか？」

「前評判は聞いていましたが，本当にここまで動くとは。感動しました」

「タケルのデータ集よりよかった」

うさ子：このようなおほめ（？）の言葉を多数いただき，スタッフの苦労も報われるというものです。CGAシステムのほうも使えるようになったら，また感想を送ってくださいね。

<質問です>

「７月号の付録ディスクには，誌面で紹介されていたRENCON.XやEXPOINT.Xなどが入っていないようですが」

うさ子：あらあら，マニュアルもよく読まないうちに，CGAシステムを展開してしまいましたね。付録ディスクから，フロッピーディスク版のCGAシステムを作ると，ディスク容量の問題ですべてのツールを収められません。

ハードディスクにインストールすれば，すべてのプログラムが使えます。ただし，ハードディスクにインストールしても，PESのメニューには表示されません。コマンドラインから起動する必要があります。

どうしてこのようにしたかというと，ハードディスクを持っていない方や，PESを使用する方は初心者なので，あまりむずかしいツールを入れても使えないんじゃないかと思ったんです。でも，コプロセッサを持っているから，RENDXVI.Xは使いたいという方もいらっしゃるかもしれませんね。

どうしても，すべてのツールをフロッピーディスクで使用したいのでしたら，新しいディスクを用意して，

LHA E DOGACGA3 B:

として，展開してください。

<クレームです>

「X68000 XVIのユーザーですが，16MHzの場合，アニメーションの画面が乱れてしまいます。10MHzのときは大丈夫でした。乱れないようにする方法はないでしょうか？」

うさ子：うーん，こちらのX68000 XVIではちゃんと動きます。そこでシャープさんに問い合わせてみました。

「実験してみたところ，確かに正常に動くX68000 XVIとそうでないX68000 XVIがあります。どうやら，初期ロットでは動かないようです。しかし，ハード的に問題になるような点は思い当たりません。原因は不明です」

うさ子：ということです。こちらではちゃんと動くので，プログラムも直しようがありません。いまのところ残念ながら，10MHzにして，我慢してご使用いただくしかなさそうです。また新たな情報を入手しましたら，誌面上でお知らせいたします。ごめんなさい。

<マニュアルについて>

「郵便受けに２つ折りで入れられないように，ちゃんと“２つ折り厳禁”と書いてくださいね」

「マニュアルは，ドキュメントファイルをつけて，ユーザーにプリントさせれば簡単では」

うさ子：甘いですね〜。あのマニュアルを２つ折りにできる郵便屋さんがいるとしたら，相当の腕力の持ち主ですよ。マニュアルは一太郎のファイルで，ディスク３枚分あります。テキストファイルで配布しても図が入らないし，プリントアウトの紙代やトナー代は１枚７円ぐらいだから，７（円）×800（ページ）＝5,600円ぐらいかな。ねっ，多少手間がかかっても，こちらで印刷したほうがいいでしょ。

<重箱の隅に間違い発見>

「CGAシステムをOh!FMにつけるとは大胆ですね。でも，うれしいです」

うさ子：「Oh!FM」は，現在「Oh!FM TOWNS」となっています(そういう問題じゃない？)。こうした突拍子もない間違いを見つけると，ついつい披露したくなる悪いクセがありますので，ご注意ください。

<肉体カンパ>

「僕は実費の2,000円だけしか送らないことにします。カンパ分は僕の兄がスタッフとして働きます。マリオ古本の弟より」

うさ子：遠慮せずに，あなたも阪大に入学してDōGAの一員となり，あなたの体でカンパをしてくださることを期待しております。受験，がんばってくださいね。

以上，ほんの一部しかご紹介できなくて残念ですが，さまざまなご意見，ご声援を多数いただきまして，本当にありがとうございました。今後とも，どんなことでも結構ですから，お便りお待ちしています。

○データ出力

概要：作ったモーションデータをセーブします。

操作：1) 「6) ファイル」を左クリックする

→FILE操作ができる状態になる

SAVE
LOAD
中　止

（「SAVE」が反転表示されている）

2) 「SAVE」を左クリックする

→ファイル出力の状態になる

どちらのファイルで出力しますか？

フレームソース
(FSC)
フレームファイル
(FRM)
中　止

（「フレームソース」が反転表示されている）

3) 「フレームソース」を左クリックする

→ファイル名入力状態になる

4) 「f1a」と入力し，リターンキーを押す

→ディスクをアクセスし，初期メニューに戻る

解説：出力ファイル名は，5文字以内なら別に何でもかまいません。フレームソースとフレームファイルの違いについては，「教養課程/CGAシステム基礎概論」をご覧ください。別にどちらで出力しても結果は同じなのですが，「AUTO」使用時に差が出ますので，とりあえずフレームソースでセーブしてください。

○FFEの終了

概要：FFEを終了します。

操作：1) 「7) 終了」を左クリックする

→画面がクリアされ，コマンドライン「B>」に戻る

○AUTOの実行

概要：いま作ったモーションデータをもとに，アニメーションを制作します。作画からアニメーションまで，すべて「AUTO」というツールが自動的に行ってくれます。作画枚数は20フレームで，作画時間はおよそ10分，X68000 XVIのコプロセッサ付きなら2,3分でしょう。

操作：1) 「B>」のあとに，キーボードから，

図8　視点設定決定

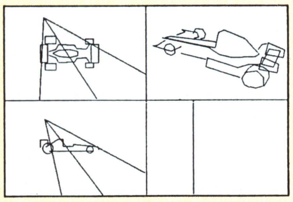


図9　軸の正負

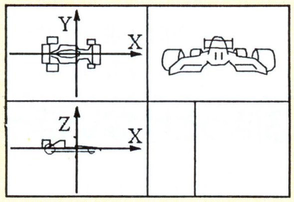
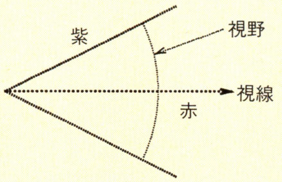


図10　視線と視野



CGAコンテスト事務局より

第5回アマチュアCGAコンテスト作品募集

○CGAコンテストとは？

もうすっかり定着した「CGAコンテスト」。といっても連載第1回目だから，ちゃんと説明しないといけないんでしょうね。

このコンテストは，“アマチュアCGA作品の発表の場を設け，質的向上を促進する”という趣旨で，毎年開催されています。毎年，数多くの傑作が発表されてます。「TORNADO」の文月さんや，「響子in CGわ〜るど」の寺尾さんも，このコンテストの出身です。

このコンテストは当チームが主催していますが，DōGA CGAシステムにこだわらず，どんな機種，どんなソフトでもかまいません。昨年のグランプリ受賞作「猿蟹合戦」はAMIGAで制作されています。ただし，単一の静止画や，プロの方が業務用の機材を用いて作った作品などは応募することはできません。

締め切りは12月31日。応募希望者は当事務局までご連絡ください。詳しい応募要項をお送りいたします。

○1カット，4カット部門新設のお知らせ

皆さんの作品発表場として始まったCGAコンテストも毎年レベルが上がり，もう常人の参加する余地がなくなりつつあるとまでいわれるようになりました。そこで，今回から新しく，1カット部門，4カット部門を設立する予定です。

1カット部門：1カット（タイトル含まず），15秒以内，BGMなし

4カット部門：4カット（タイトル含まず），30秒以内

1カット部門は作品性などもあまり問わず，とりあえずこんなカットを作ってみたから見てほしいというような初心者のための部門です。また，4カット部門は一般部門に応募するほどのパワーはないが，CGAコンテストには参加してみたいという方のための部門です。4カットあれば作品性も出せますし，うまくすれば映像短歌というべき面白いジャンルになるのではないでしょうか。

この2部門についてはあまり厳しい審査はせず，できるだけ多くの方の作品を入選させる方針です。そして，コンテストの上映会やビデオの中で，ラッシュフィルム的に上映します。この部門は賞金が出るわけではありませんが，発表の場として，積極的に活用してください。

応募方法は基本的に通常の作品と同じですが，ビデオテープではなくフロッピーディスクで送ってください。

コンテストビデオ配布終了のお知らせ

○ビデオはちゃんと届きました？

遅れに遅れた「第4回CGAコンテスト入選作品集」ビデオの配布が終了しました。待ちに待たれた皆さん，ごめんなさい。CGAシステムのマニュアルを申し込んだ方はマニュアルを見て，遅れたわけを納得してください。

さて，以前「愚か者め」のコーナーで指名手配したなかで，まだ連絡がない方がいらっしゃいます。とにかく，申し込んだのにまだ届いていないという方がいらっしゃいましたら，至急プロジェクトルーム内「ビデオが届かん，なぜだろう係」（担当：マリオ古本，遊び人松井）までご連絡ください。

○最後の愚か者め

一．4月21日に新宿郵便局から申し込んだ，名なし住所なし

二．5月18日に広島郵政研修所前から申し込んだ，名なし住所なし

三．5月29日に電信振り込みにて3,000円振り込んだあと，加入者負担で振り込んだことに気がついて，6月1日になってその分の210円をわざわざ振り込んだのはいいけど，両方とも住所を書いていないヤマグチヒロシさん

○宛名，梱包作業要員募集中

なお，コンテストのビデオ配布サービスは，当チームにとって大きな負担となるので，今回をもって終了させていただきます。といっても，みんな許してくれないだろうなあということで，来年の配布に向けて，宛名書き，梱包作業要員を募集します。

なお，アルバイト料，交通費などいっさい出ません。プロジェクトルームへの交通の便のいい方，ご連絡ください。

auto f1a.fsc

と入力し，リターンキーを押す

→いろいろ表示されるが，無視しておくと10分後にアニメーションが始まる

2)　「ESC」キーを押して終了する

解説：作画時間は，コプロの有無，周波数（10MHz/16MHz）によって異なります。このままでは画質がかなり悪いですが，画質の改善方法はあとですぐ解説しますのでご安心を。逆に，モーションデザインのチェックだけをする場合，ワイヤーフレームでもっと高速に作画させることもできます。

wireview /v f1a.frm ＊.suf

とすると，30秒程度でアニメーションを表示します。

○画質の改善

概要：「AUTO」実行時にオプションをつけ，画質をよくしてアニメーションさせます。作画時間は50分，X68000 XVIのコプロ付きなら10分程度です。

操作：1)　「B>」のあとに，キーボードから，

auto f1a.fsc　/a3　/g　/d

と入力し，リターンキーを押す

→基本的に，「AUTOの実行」と同様だが，作画が終わったときに，「動画F1Aについて，出力ファイル名は？」と表示されて止まる。この場合，何も考えずに，リターンキーを押せばよい

解説：オプションについて

/a3　画質をよくするオプション

輪郭のギザギザがぼやけて少なくなる。理論的には，「/a」の後ろの数字が大きいと作画スピードが遅く，その分画質がいいはずだが，実際は4以上にしても画質の差はない。だから，通常は2か3にする。

/g　曲面にするオプション

曲面も直面の集まりだが，直面の継ぎ目の色の変化をなくして，なめらかな曲面のように見せる。とはいっても，全部の面が曲面になるわけではなく，形状デザイン時に，あらかじめ指定しておかなくてはいけない。なお，マッピングをするときも，このオプションをつける。

/d　色数を減らすオプション

“/a3”や“/g”オプションをつけると，1画面中の色数がどんどん多くなってしまう。アニメーション実行時の色数には制限がある（制限を外す方法もある）ので，アニメーションを実行する前に，色数を減らすプログラム「CRD」を実行する。

ほかに512×512の解像度のアニメーションもありますが，スピード（1秒間の枚数）が遅くなるので，やめておいたほうがよいでしょう。

また，画質が不満だという方もいらっしゃるでしょうが，最終的にVTRに録画することを考えると，これ以上の画質はあまり意味がありません。

おわりに

いかがでした？　操作手順どおり実行するのは，結構面倒だと思いますが，ほとんどの操作が見ればわかる程度のことなので，慣れればこの程度のアニメーションはサクサク作れるでしょう。

今回制作したアニメーションを見て，“「お試しシステム」と変わらないじゃないか”と思ったかもしれませんが，連載もまだ1回目ということでお許しください。

とはいっても，「お試しシステム」とは大きく違う点があります。それは，すでにあるデータを流用したのではなく，モーションデザインを自分で行ったという点です。“CGAシステムなんて，自分にも使えるんだろうか”と不安に思っていた方も，安心されたでしょう。

来月はFFEをさらに使い込んで背景をつけ，複数の車を走らせます。今月解説したFFEの使い方は完全にマスターしておいてください。

さて，たくさんのマニュアルの申し込みをいただき，誠にありがとうございました。在庫の山を抱えるとか，発送処理が追いつかないといったトラブルもなく，現在のところ順調に発送されています。正しく申し込んだ方は，遅くても9月初めまでには到着すると思います。正しく申し込んでいない方は，それなりに遅れます。

申し込み用紙の自由記入欄にご意見，ご感想をぎっしりと書いてくださった皆さん，ありがとうございます。“どうせこんなところは読んでないだろうが……”と書いていた方もいらっしゃいますが，私やうさ子をはじめ，数人のスタッフはすべての申し込み用紙に目を通しています。よく読んで今後の参考にさせていただきます。

CGAマガジン編集部より

DōGA CGAマガジン（仮称）とは，皆さんから送られたデータをディスクにまとめ，マウスひとつで作画，アニメーションを実行できるようにしたデータ集です。

すでに，

「ディスクマガジンに非常に期待している」

「ディスクマガジンについての詳しい情報を」

「送りたいデータがあるが，どうすればよいのか？」

といった問い合わせを多くいただいています。

発表方法など，詳しいことはまだ決まっておりません。とりあえず，年内発表を目標にシステムを開発している段階です。第一，皆さんからデータが送られてこなければ，発表しようがありません。

ということで，まず，データを募集します。

・形状データ，フレームソースと，実行用のバッチファイル

・形状データだけでもいい（こちらで適当に動きをつけます）

・アトリビュート名などはちゃんと整理するか，詳しいドキュメントをつけてください（こちらで修正するときに必要）

・著作権に引っかかるようなものは不可

作品はフロッピーディスクに収めて，下記の宛先まで宅配便か，普通郵便で送ってください。フロッピーディスクのケースに入れて発送すれば，途中で壊れることはまずないでしょう。

なお，賞金などはいっさいありません。また，データ集として多くの方に自由に使ってもらうため，著作権を放棄，あるいはDōGAに委任していただく必要があります。あらかじめ，ご了承ください。

P.S.　大石さん，“第2次世界大戦機シリーズ”は任せました。

○応募先

〒533　大阪市東淀川区淡路5－17－2　102号

プロジェクトチームDōGA「CGAマガジン編集部」

Communication SX-68K, そしてFIXER

Ogikubo Kei 荻窪 圭

SX-WINDOWにもようやくアプリケーションが揃ってきました。そして，今度は通信ソフトが登場。パソコン通信には不可決なものだけど，使い勝手はどうでしょうか。新しいFIXERもちらりと紹介します。

とうとう入ってしまったのだよ。NTTのテレジョーズとかいうアヤしげなやつに。NTTもいろいろと考えるものだ。

しかし，そもそもNTTの料金体系で許せないのは，隣接区域，そのまた隣接区域と，指数関数的に高くなっていくことで，たとえば市内だと3分10円だけど，その隣になると，10円で半分以下しかかけられず，さらにその隣になると，38秒だか30秒だかで10円という法外さ。これがどんどん進んで，最後には4.5秒で10円だかになる。これはおそろしいことである。

宅急便だって（どう考えても，電話回線内を音声が流れていくより，車に荷物積んで走ったほうが，距離の影響は出るはずなのだが）ここまであこぎな料金体系ではない。そもそも，あの料金体系は交換機が手動でいろんな人の手をわずらわさねば遠くへつなぐことができない時代の名残であって，自動化された現在，これだけ価格差をつける意味がどこにあるのか。

値下げ，値下げと声を高らかに，深夜がどうとか，遠隔地がどうとかといっているが，あんなもん小手先である。

数年前，それまで0423という市外局番のところに住んでいた私は03っていう市外局番のところへ引っ越した。電車でほんの15～20分ほどの距離だ。なのに，月々の電話代は確実に1/3に減った。前はちょっと通信しただけで，簡単に20,000円とか30,000円になっていたのに，いまは毎日通信しても10,000円前後ですんでいる。電話を使う時間はむしろ増えているのに，だ。

効果が大きそうに見える遠距離に気をとられて，近距離市外通話を野放しにしてはいけないのだ。

そういうわけで，話は戻ってテレジョーズとやらである。ずいぶん前からやっているので，知っている人も多いかと思うが，とりあえず説明する。

2,000円，3,000円，4,000円，5,000円の4コースがあり，5,000円コースだとボトル1本サービス……ちゃうちゃう。えっと，4つのコースがある。たとえば，2,000円コースだと自動的に250円割り引きになる。さらに，2,000円を超えた場合，3,000円までの1,000円分に対して，15パーセントの割り引きになる。

じゃあ，なんで2,000円コースかっていうと，電話をしようがしまいが1,750円とられる（割り引かれた250円を加えて2,000円ね）からである。8,000円コースだと，自動的に1,150円割り引きになる。が，電話をまったくしなかったとしても，8,000−1,150＝6,850円は必ずとられるのである。1カ月期限のプリペイド方式なわけだ。毎月絶対に6,850円以上電話するって人以外は得をしないのだ。

しかも，夜10時から翌朝8時だけが対象。コンピュータ化したとはいえ，なんと面倒なシステム。テレジョーズがなかったらNTTに入るはずだった総金額（割り引き分）と，テレジョーズにしてしまって余分に払った総金額（電話料金が定額に満たなかった分）の差額を知りたいものである。後者のほうが多かったら，許せんぞ。

さて，Communication SX-68K

という感じで，Communication SX-68Kの話へ入っていくわけだが，その前に，通信ソフトには2種類のコンセプトがあることをおさえておきたい。特に，電話回線を使ってネットワークシステムにアクセスすることを主目的とした（UNIXの端末になるためのソフトではないということだ）通信ソフトに対して，だ。

1）オンラインの状態を快適に過ごすためのソフト

2）ログをあとで読み返す人のためのソフトである

たとえば，前者であれば，広大なバックスクロールバッファやレスポンスの高速性，簡単な操作が重要になるが，自動運転のためのマクロはそれほど高度でなくてかまわない。あえて，こちらをオンライン指向といおう。

後者であれば，バックスクロールバッファやレスポンスなどよりも自動運転のためのマクロが重要になる。マクロが優秀であればあるほどユーザーはアクセス操作をパソコンに任せ，通信終了後，エディタなどでゆっくり処理をすればいいわけだ。こちらをバッチ処理指向と呼ぶことにする。

オンライン指向の例がMuTerm。Telecom-MIKIもそうだと思う（ちょっと触っただけだから自信はないけど）。迅速な反応。バックスクロールバッファを見ながらの通信。メモリにもよるが，巨大なバッファ（私は512Kバイト確保している）。ファンクションキーの文字列登録。オートログインしか考えられていないマクロ。バックスクロールバッファから文字を切り出して送信。通信しながらのチャイルドプロセス。

一方，バッチ処理指向の例がCommunication PRO-68K。こいつでマクロを組んで，AUTOEXEC.BATで自動実行するようにし，X68000のタイマー機能を使って，“朝，自動的にX68000が起動して，必要なネットすべてにアクセスし，未読のメールをすべてハードディスクに落とす”という自動アクセスをする人さえいるという。あとは帰宅してからゆっくりとそれを読み，いくつかメッセージを書き，アップロードして寝る。翌朝，またX68000が勝手に起きて，勝手にアクセスし勝手にダウンロードしておいてくれる，のだそうだ。自分でや

っているわけではないから詳しくは知らないが，それもまたX68000らしくて面白い。

どっちがいいかはその人の通信のやり方の問題であって，とやかくいう気はない。

私は完全にオンライン指向。ログは（自分がモデレータやっているところ，必要なもの，貴重なものしか）取らないし，文章はたいていオンラインで書くし，なにより面倒なマクロを組むのがきらいだ。

オンライン指向型の通信ソフトなら，必要なログはバックスクロールバッファから切り出してそこだけ保存すればいいし，数日分なら保存しなくてもバッファに残っていてくれる（あ，もちろん，X68000の電源は入れ放しにしなければならないが）。さらに，指定したファイルをバックスクロールバッファにロードすることもできるからこりゃ便利。

アバウトな性格でログを取る習慣がなくて，市内通話でアクセスできて，十分高速なタイピングが可能なら，オンラインで遊んだほうが楽しいのではないだろうか。

しかし考えてみると，そういう人ってビジネスソフトを使いこなすのに，いちばん向かない性格なんだよな。ほんと。

向く性格ってのは，毎日ログを落とし，ログ整理ツールかエディタのマクロか何かでログをきれいに（残す必要のないところを削除するなど），できる人である。

さらに凝る人は，F-CARD GTでも使って，見出しとキーワードを入れ，必要に応じて必要なログがすぐに検索できるようにする。うーん。うらやましいものだ。

ウィンドウ時代の通信

では，このSX-WINDOW初の市販通信ソフトは，オンライン指向かバッチ処理指向か，どちらなのか。

まず，Communication PRO-68Kをベースにしていること。これを忘れてはならない。あくまでも，バッチ処理指向のCommunication PRO-68Kをベースに，オンライン指向のSX-WINDOWにもってきたのだ。かなり頑張っているし，マクロも充実しているのだが，結果として，どちらへも手を出した格好になってしまった感は否めない。

オンライン指向のSX-WINDOW。

確かにそうである。アバウトにログインしてアバウトにログアウトする人にとって非常においしいのが，ウィンドウシステムの通信ソフトだ。つながったらおもむろに会議室を徘徊する。読みたいものを読み，書きたいものを書く。他人の引用をしたくなったら，引用元のメッセージへ戻って必要なところをコピーし，ペーストする。ログアウトする。重要なところだけカット＆ペーストでエディタへもっていく。

でもって，通信ソフトは常時背面に待機していて，じゃまなときはちょっと画面の外へ出てもらって，うーん，いいねえ。

さらに，電子辞書かなんかあって，ちょっとインテリジェントな書き込みしたいときは，しゅるしゅるとCD-ROMでも回してちょいと引用。それから，だだだーっとフリーウェアとか画像データとかをダウンしながら，横に開いたエディタでちょいとアップする文章を作ったりして。

甘くはないぞ現実は

欠点はどうだろうか。

Communication SX-68Kが悪いのではない。ほんと，そうだけど，なんというか，やはりスクロールなどが遅い。これはSX-WINDOWに限らず，MacintoshでもWindowsでもそうなのだけれど，オンライン指向の人が自在にネット上で遊ぶにはちょっと反応が悪い。バックスクロールバッファ上を自在に飛び回るにはつらい。

続いて，SX-WINDOWの問題なんだけど，文字が読みづらい。フォントの問題とかではなくて，色の問題だ。

あのグレイスケール4階調ってのはよくない。特にアクティブウィンドウの背景が，いちばん明るい階調ではなく，2番目のやつ，ってのもよくない。

渋いのもいいけど，通信ってのはCRTというはなはだ解像度の粗い面に表示された粗い文字を，必死に読むことを義務づけられるわけで，背景と文字のコントラストが低いと目が疲れる。あまり高くても疲れるのだけど，とにかく疲れるのだ。

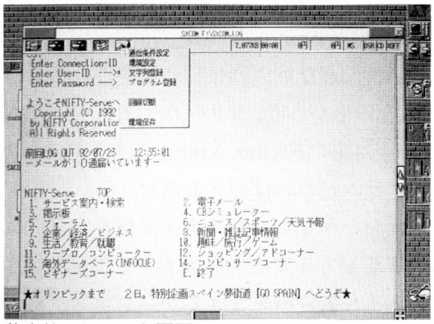
そーいえば，私は最近，顕著にCRTと目の同期がとれなくなって困っている。疲れ目かなあ。チラチラする，とかではなくて，形容しがたいのだが，CRTがひどく他人行儀に見えるのだ。なんんか，自分の目で見ているのではないみたいに。

これは，パソコンに限った問題ではなくて，テレビを見ていてもそうなのだけれど，どうしてだろう。先ほど述べたように，SX-WINDOWがことさら疲れるように思えるのもそのせいだろうか。とりあえず，いつもより部屋の明かりをワンランク上げてみたりするのだが，だめ。どうしてでしょうねえ。医者に行って，「あなたの目は劣化しています。24kHz以下のディスプレイしか見てはいけません」なんていわれたりしたら，どうしよう。X68000もMacintoshもPC/AT互換機もだめじゃないか。

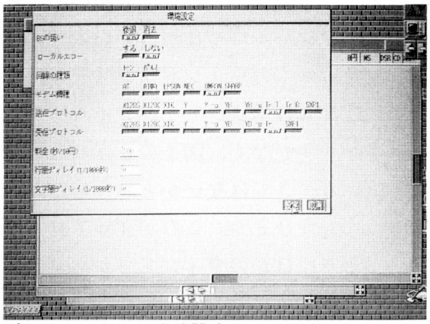
はっは。また話が逸れた。

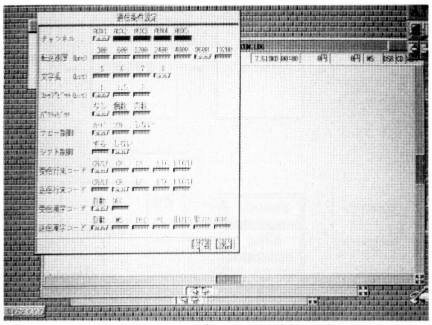
えっと，Communication SX-68Kだ。

こいつは，SX-WINDOW上のソフトである。当たり前か。シムアースと違い，2Mバイトでもいくつかのドライバを外すことなく動く。これも当たり前だ。ウィンド



基本的にはこんな画面


プロトコルなどの環境設定ウィンドウ


自動運転の編集画面はダイアログなのだ

ウの大きさも自在。これも当たり前。

いちばん“らしい”のが，文字の大きさ切り替えだな。エディタと同様，12/16/24ドットから選べる。24ドットでログを読むと不気味だぞ。なんか，異形のものを見ているようだ。ISHファイルを無手順で落として24ドットで見ていたりすると，トリップするかもしれない。

24ドット表示にすると，当然横80桁は表示できないわけで，横にはみ出る。もちろん，スクロールオンで使うこと。12ドット表示ができるのは（目は痛くなるけど）うれしい。大量に表示できるのは願ってもないことだ。

それでは，具体的なところを順番に見ていこう。

タイトルバーの下にはポップアップメニューが5つ並ぶ。その右には通信ソフトにありがちな各種情報。電話料金とか接続時間とかディスクの残り容量とか。

ポップアップメニューは左から，電話アイコン，ダウンロードアイコン，アップロードアイコン，プロトコル通信アイコン，環境設定アイコンと，オーソドックスな作りだ。

ちなみに，Macintosh用のJtermという和製市販通信ソフトでのメニューは，“ファイル”“編集”“設定”“ダウンロード”“アップロード”“エディタ”となっており，通信ウィンドウについているボタンは，“ダイヤル”“ログイン”“ブレーク”“オフライン”“マクロ”である。Communication SX-68Kと似ているでしょう。

これが，英語版をローカライズしたシェアウェアのTerminal 2.1Jだと，“File”“Edit”“Option”“Script”“Macro”と，アップロードとかダウンロードというメニューはなくなる。これらはファイルメニューに“Save Buffer”とか“Send Text”などという名前で入っているわけだ。

面白いのは，日本のソフトの多くが，ダウンロード，アップロード，プロトコル通信，ってのをちゃんと独立メニューにしているところだ。なにか特別な操作であるかのように。

電話をかける

さあ，電話アイコンである。ここは“自動ログイン”“自動運転実行”“自動運転強制終了”の3つが埋まっている。

自動ログインと自動運転ではどこがどう違うんだ，などとはいわないように。マニュアルが刷り上がってないので確認していないが，自動ログインというのはソフトに登録したホストへのログインを，自動運転というのはエディタで組んだマクロを実行するためにあるのだと思われる。

この自動ログインを実行すると登録してあるホストのメニューが現れるわけで，選んで“編集”ボタンを押せば自動ログインを実行するための設定画面になる。これがちょっと極悪なので文句をいっておく。

まず，通信条件設定である（環境設定メニューで出てくるやつと一緒）。これは問題ない。ここで“設定”をクリックする。続いて，環境設定である。これも環境設定メニューで出てくるやつと一緒。設定したら“設定”ボタンをクリック。続いて，自動ログイン設定である。ここではホスト名，同時に実行するマクロ名，電話番号，自動ログインのためのプロンプトと送信文字列を入れる。いちいちエディタで書くのに比べたら楽である。

これで終了。

何の問題もないように思えるでしょ。しかしである。文句が2つ。

ひとつは，それぞれの画面がシーケンシャルにつながっていること。せっかくのウィンドウシステムなのだから日本式シーケンシャル的フレンドリでなく，図1のようになっているべきではないかと思うのだ。

もうひとつ。これがダイアログであり，ほかのウィンドウからコピーしてもってくる，ってことができないこと。

だって，そうでしょ。これを買う大部分の人は，いままでHuman68k上でなんらかの通信ソフトを使っていて，そこには自動ログイン用の設定ファイルがあるはずである。ということは，電話番号やらプロンプトやらIDやらを，またいちから打ち込み直すのは面倒臭いではないか。

あらかじめコピーしておけば，1文字列だけはもってくることができるが，それだけ。さあ，設定しよう，と思った私はそこでくじけたのである。

ここで指定したログイン手順は，プログラム名で指定した名前のテキストファイルにマクロとして生成されるため，ほかのソフト用のログインマクロを持っている人はとりあえず適当に作っておいて，あとからエディタでコピー&ペーストして作ったほうが楽だ。

ダウンロードとアップロード

Communication PRO-68Kの流れを継ぐものだけに，なにやら面妖な言葉づかいがなされている。ダウンロードとはいわず，ログファイル保存という。アップロードとはいわず，オートタイプという。

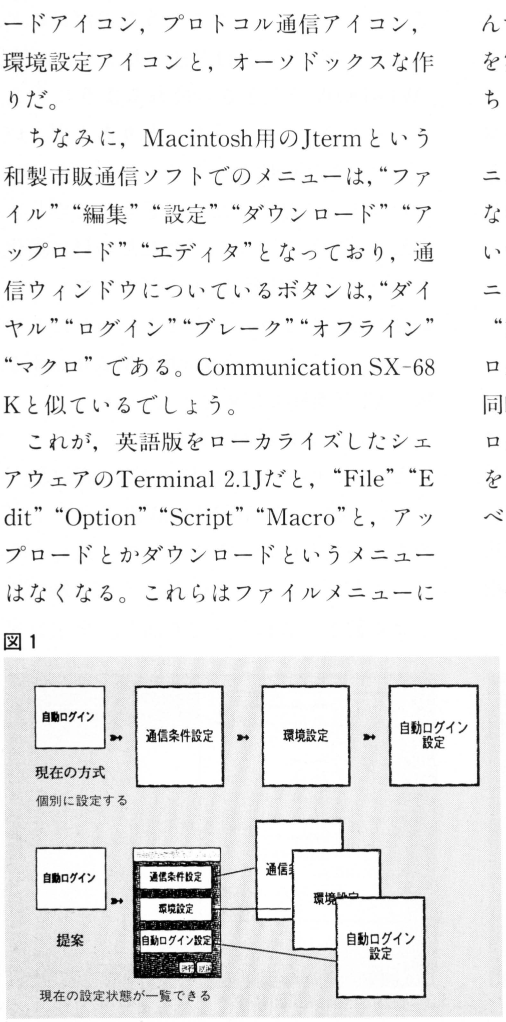
まあ，いいのだが，個人的にはそうだなあ，ログ保存ってのはいいとして，オートタイプは気に入らない。テキストファイル送信くらいがいい。

で，プロトコル送受信（バイナリファイル送受信となっている）であるが，対応しているのはXmodem，Ymodem，TransIt，SXP（なんじゃこれ？）となっている。足りない。BPlusなど大手ネットワーク標準プロトコルくらいは全部サポートしてもらいたい。NIFTY-Serveをよく使う私としては，まずBPlusである。ついでに，Zmodemもほしいところだ。

まあ，送受信に関しては，BPlus（NIFTY-Serve，CompuServe）とQuick VAN（PC-VAN）がない以外は特に問題ないでしょう。なんというか，フリーウェアでSX-WINDOW対応のFlying XmodemとかBplusプログラムがあれば補える話だ。

欲をいうなら，バイナリ受信を自動的にやってもらえるとありがたい。Bplusだとファイル名も指定できるから，ホスト側がバイナリ送信の準備をしたとたんにそれを

図1



検知してバイナリ受信モードに入る，って機能があると，非常に楽だ。現にそういった機能は，Macintosh用ではすでに一般的である。

ポップアップメニュー

SX-WINDOWの場合，ポップアップメニューが使い勝手を決める。通信画面でのポップアップメニューは“カット”“コピー”“ペースト”“消去”“送信”“保存”“全選択”“前方検索”“後方検索”と必要なものはひととおり揃っており，何の問題もない。通信ソフトでログから“カット”できるやつ，ってのはこれが初めてだが，まったくもって悪くない。面白い。

バックスクロールバッファも（マニュアルがないので最大値はわからないが），少なくとも4000行くらいはお茶のこさいさいだった。つまり，でででーっと適当に通信して，その場でいらないところをカットして，保存する，ってことが簡単にできるわけで，なかなかグレートである。

さらに，SHIFTを押しながらポップアップメニューで，10個まで登録できる文字列の送信が，CTRLを押しながらポップアップメニューで，登録したプログラムが，OPT.1を押しながらポップアップメニューで，自動運転終了や回線切断やブレーク送信など強制終了関係のメニューが現れるようになっており，これもまたよいことだ。

通信画面での心残りは，ウィンドウの縦分割ができないことくらいだ。縦分割っていうのは，ウィンドウを縦に分割して，上にはログから必要なところを出しておき，下で通信する，っていうMuTermなどが得意としている技。100行前のメッセージへのコメントをオンラインで書きたいとき，などに便利だし，さらに，まとめてダウンロードしているときに，終わった分をゆっくり読むにも最適だ。できたら，そのうち，つけてもらいたい。オンライン指向ユーザーには欠かせない機能だ。

それでもなかなかよい

なんだかんだいっても，スクロールが遅いってのはあるが（いまのバージョンではエディタよりも遅い），悪くないソフトである。空いているところにエディタを開いておいて，そこにまとめていくつもコメントを書いておき，必要に応じてコピー&送信してやれば非常に楽だし，なかなかウィンドウシステムしていてよい。あまりにオーソドックスでつまんないけど，変に奇をてらわれるよりはよほどよい。いくつかバグもあったが，製品版では直っているはずだと思う。

オーソドックスでよかったね，って感じだ。ああ，メモリを増やさねば。

話題変わってこちらはFIXER

さて，ここで話題が変わって，FIXERである。皆様待望（皆様かどうかは知らぬが），FIXERがバージョンアップしました。FIXER.SYSのバージョンは1.20。うれしいねえ。ああ，うれしい。ほんとにうれしい。

何がどう変わったか。とりあえず目につくところでいくと，

1) キーコンフィギュレーションが可能
2) WP.Xに対応
3) SX-WINDOWへの配慮（高速化）

ってところだ。どれも目玉だね。WP.Xに対応，ってことで期待したBuisiness PRO-68K popularへの対応であるが，なんとか動いてちゃんと変換できた。いいことである。これで普通のユーザーにもFIXERが安心して使える状況がやってきたわけだ。

ただ，WP.XとBuisiness PRO-68K popularのとき，アプリケーションを終了しても全角キーとローマ字キーがついたままだというのは気になったが，動いただけよし，だ。もう，FIXER得意のCTRL＋XF4でローマ字ON/OFF，っていうような技はきかないけど，WP.XやBuisiness PRO-68K popularで動くのはめでたしである。

さて，1)のキーコンフィギュレーション。こいつはASK 2.0のものに準じているようだ。なかなかよろしい。

FIXERってのは，知らない人がいるかもしれないからいっておくと，PC-9801用のFIXER ver.4.0のX68000版。いまはWORDYってワープロソフトにバンドルされている。

こいつの特長は，AIだかなんだか知らないけど，きめ細かい品詞情報と学習機能の賢さ。本当に賢い。文節学習をちゃんとしてくれるので，ヘンなところで切って変換したケースでも再現できるのだ。

さらに，さっきもちょっと触れたけど，CTRL＋XF4でローマ字ON/OFF，CTRL XF5でCAPS ON/OFFってのもうれしい。私は個人的にいちばん好きな日本語FEPである。ASKよりもEG BridgeよりもATOK7よりもVJE βやγよりも好きである。WXII＋やKatanaより好きか，っていわれると，この2つは使ったことないから自信がない。

辞書は開くもの

FIXERはいい。なにより変換効率である。学習するのである。辞書ファイルが大きいわりに登録単語数はいまいちだが，そんなものは自分で登録すればいい。

ユーティリティ（FIXUT.X）のメニューから「ユーザー登録熟語の一覧」ってやつを使うと，辞書をテキストファイルに変換して吐き出してくれる。これを参照しながら登録熟語を書き出して，「辞書の合成・編集」を使えば簡単にできる。

これからの時代，キーを握るのは，閉じた世界で優れたものではなく，優れていてなおかつ開かれたものだ。辞書データだって同じこと。本当は，X68000ユーザーの100パーセントが持っているASKっていう代物があるのだから，ASKのユーザー登録辞書から必要なものをFIXERの辞書に転送するようなユーティリティがあっても面白かったかもしれない。

で，問題は，MuTerm&FIXERの軽量軽快イケイケコンビから，SX-WINDOW&FIXERへ通信環境をリプレースするか，である。むずかしいなあ。MuTermのそれなりにまとまっていて，それなりに完成度が高くて，高速なのもいいけれど，SX-WINDOW環境の，エディタをいっぱい開いておいて，コピー&ペーストしまくりながら通信する，ってのにも惹かれる。しかし，電話代を気にするなら，MuTermだな。まあ，すべては製品版が出てからである。

あとはテレビでも見ながらゆっくり考えるとしよう。ええい，ナイターなんてやってやがる。誰か巨人を定位置（最下位のことだぜ）に戻してくれんか。これではおちおち眠れない。もう中日はあきらめた。さあ，頑張れヤクルト！

★(影)のショートプロぱーてぃ その36

書は心を表すものなり

Kageyama Hiroaki 影山 裕昭

もう秋なんですねぇ〜。食欲の秋，読書の秋といろいろあれど，やっぱり秋といえば芸術でしょう。ということで，今月は日本の伝統美「書道」の登場です。ほかに疑似的にワイルドカードを使えるようにする「WILD.C」を紹介します。

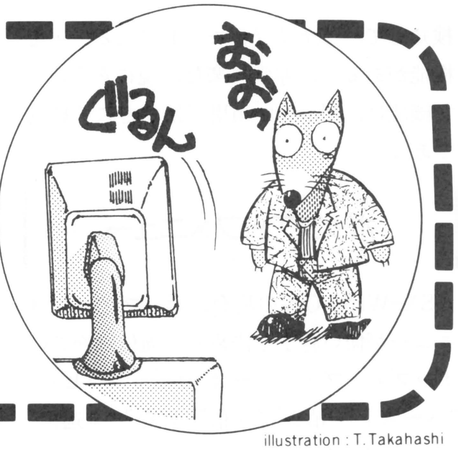
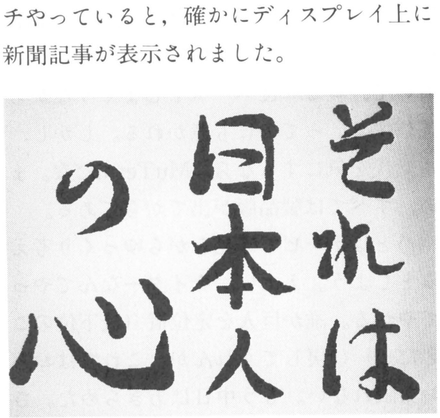


illustration : T.Takahashi

先日，新川崎にある“日立システムプラザ新川崎”に行ってきました。そこは地上31階，地下2階のツインビルディングで，まるでX68000のツインタワーを彷彿させるような建物。中に入ってみるとインテリジェントビルといわれるだけあって，埼玉の田舎に住んでいる私には縁のないハイテク空間が広がっていました。ビル内の各フロアは高速光LANで結ばれていて電子メールで連絡はできるし，電話回線を使ってのテレビ会議はあるし，子供の頃に本で見た「未来の日本」が現実のものになっているんですねぇ。その日は珍しいものをたくさん見せてもらったんだけど，その中に思わず「オオッ」と声まであげて驚いてしまったものがひとつあるんですよ。それは，説明のお姉さんに連れられて図書室の部屋の隅に置いてあった1台の端末（日立製）の前へ案内されたときのこと。

「ペーパレス化を図るため新聞の記事はスキャナで取り込み，光ディスクにファイルされています。簡単な操作で指定した日時の新聞記事をディスプレイに表示することができます」

説明しながらお姉さんがマウスをカチカチやっていると，確かにディスプレイ上に新聞記事が表示されました。



SHODO68K.BAS

「文字が横に寝ていて読みづらいときは，画面を90度回転することもできます（ファンクションキーを叩く。ウィィーン）はい，このようになります」と営業スマイルのお姉さん。

なんと！ 縦置きだったディスプレイが，ディスプレイごとグルンと90度回転して横置きになってしまったのです。これにはビックリしましたよ。まさかハードウェアでディスプレイごと回転させてしまう荒技が存在するとは。それにしても，手で回せばすむようなものをわざわざ作ってしまう技術屋の人っていったい……。

サイバー書道なのぢゃ

今月の1本目は，X68000のディスプレイを半紙，マウスを筆代わりにして書道をやってしまおうというプログラムです。

SHODO68K.BAS for X68000
(要APIC.FNC，X-BASIC)
東京都 坂本 康

さて，まずはAPIC.FNCを使うための準備をしましょう。APIC.FNCは創刊10周年記念PRO-68Kに収録されています。それをBASIC.Xと同じディレクトリにコピーしておいてください。次にBASIC.CNFに，

FUNC = APIC

を追加してください。これで準備は完了です。

X-BASICからリスト1を入力してRUNで実行してください。画面が真っ白になりますね。画面をジーッと見てください。ほうら，だんだん半紙に見えてくる，見えてくる。はい，これは誰が見ても半紙ですね。マウスカーソルを動かして筆を下ろしたい位置までもってきたら，おもむろに左クリックしてください。すると，マウスカーソルが緑の円に変わります。そして，そのままマウスを動かすと，円の太さで線が書かれます。

しかし悲しいかな，インタプリタでは処理速度が遅いため，あまり速くマウスカーソルを動かすと，線にならずに円がポツポツと描かれます。16MHzのX68000なら書道できる速度ですが，10MHzだと線を書くのが難しくなり，書道以前の問題です。コンパイラをもっているなら，後述する方法でプログラムをコンパイルして実行してみてください。そうでない方は……素直に諦めてください。

マウスを動かしながら左ボタンを押し続けると，線が太くなっていきます。右ボタンを押すと線が細くなっていき，最後には元のマウスカーソルに戻ります。つまりマウスの両ボタンで筆圧を調節するのです。マウスカーソルが表示されているときは筆を持ち上げている状態というわけですね。

画面の左上にはなにやら怪しげな記号があります。ここをマウスでクリックしてもなんにもなりませんよ。マウスカーソルを画面中央に移動してから，ROLLUP，ROLL DOWNキーを押してみてください。画面が上下にスクロールしますね。それにつれて左上の表示も変化します。“・”が半紙全体の大きさで“I”が現在表示されている部分なのですね。こんなとこまで作ってしまうなんてスゴイ凝りようですねぇ。ちなみに投稿原稿によれば，「縦2画面分なのでお正月の書き初めなどはできません」だそうです(ってX68000でそんなもんするかっ)。

これだけではありません。ESCキーを押すとメニューが表示されちゃいます。ここでセーブを選択すれば，書いた作品をディ

スクに保存することができます。ファイルネームを聞いてきますので，拡張子を付けずに入力してください（拡張子は自動的に.SHOが付きます)。作品は実画面1024×1024,16色モードで(0，0)－(512,1024)の範囲がAPIC形式でセーブされます。

うーん，このプログラムはアイデアの勝利という感じですねぇ。書道というと線の太さで右はらいや止めといったものを表現するわけですが，それがうまくできないとどんなに文字のバランスがよくても見栄えがよくないんですよね。SHODO68Kでは線の太さをマウスボタンで調節することによって，本物の書道と同じような文字を（訓練しだいで）書くことができるんですからたいしたもんです。ディスプレイからあの独特の墨の匂いがしてきそうですよ。書いた作品を保存できるという点も見逃せないですね。

投稿原稿によると，X-BASICにCIRCLE FILLコマンドがないので苦労したようですね。結局，LINEコマンドでCIRCLE FILLと同じ機能の関数を作ったんですね(200～300行)。

これがX68000での初めてのプログラムだというのですから，恐れ入ります。

まさかこれで書道の宿題をやったりすることはできないし，実用性は乏しいかもしれないけど，こういうプログラム，私は大好きです。ショートプロなんだもの，実用的なのもいいけど，こういうお遊びプログラムがあってもいいんじゃない。

そうだ，忘れるとこだったけどコンパイルの手順を説明しましょう。最初にSHODO68K.BASを1カ所変更します。

```
680 apic_load(fn,0,0)
```

としてください。次に創刊10周年記念PRO-68Kに収録されたAPIC_LIB.AをSHODO68K.BASと同じディレクトリにコピーしたら，

```
A>CC /W SHODO68K.BAS APIC_LIB.A
```

でコンパイルすることができます。

ワイルドなコマンド

さて，次のプログラムはワイルドカードの使えないコマンドで，疑似的にワイルドカードを使えるようにしてしまおう，というものです。

WILD.C for X68000

（要Cコンパイラ）

大阪府　野崎哲也

実はこれ，最初はWC.Cというファイル名だったんだけど，WCというのはUNIXで存在するコマンドなんです。すでにX68000にも同名のコマンドが移植されて出回っているということなので，こちらでWILD.Xに変更させてもらいました。野崎さん，そういうことなのでご了承ください。

さて，ワイルドカードというのは“＊”と“？”のことです。“＊”で任意の文字列を表し，“？”で任意の半角1文字を表します。たとえば，Z-MUSIC用の演奏データのファイルを表示しようと思ったら，

```
A>DIR ＊.ZMS
```

とすれば，拡張子が“.ZMS”のファイルを表示することができます。このようにワイルドカードを使うと，目を皿のようにしなくても特定ファイルを検索することができるんです。

しかし，この便利なワイルドカードも，使えるコマンドが限られているんですよねぇ。たとえば，ZMUSIC.Xではワイルドカードが使えません。ZP.Xでジュークボックス演奏させるために，複数のZMSファイルをコンパイルしたいときは，いちいち，

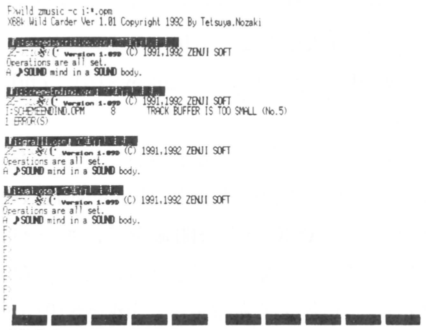
```
A>ZMUSIC －C ABC.ZMS
A>ZMUSIC －C 123.ZMS
        ⋮
```

のように同じような入力を繰り返さなければいけないのです。これがまた面倒な作業なんですよね。そんなときにWILD.X。これさえあれば，

```
A>WILD ZMUSIC－C ＊.ZMS
```

とするだけで，自動的に拡張子が.ZMSのファイルを取り出して順次コンパイルしていってくれるのです。いままでの苦労はなんだったんだ～。

なお，上の例のようにWILDのあとにくる実行ファイルがオプションスイッチを取る場合は，“/”，“－”をスイッチとみなし，それ



WILD.C

以外のいちばん最後のものをファイル名とみなします。投稿原稿には「WILD.Xは取り出したファイル名を黄色のリバースで表示してから実行するので，もともとワイルドカードに対応している「TYPE」コマンドでも，ファイルの変わり目がすぐわかるので便利です」ということ。うんうん，ホントにすごい便利ですよ，このプログラムは。

余談ですが，実は同機能のプログラムを瀧君が作っていたんだけど，ある程度使える段階になって未完成のまま途中で制作をやめていたんですよ。本来なら今月か来月あたりに瀧君のプログラムが掲載されるはずだったんだけど，それも夢と終わってしまいました。瀧君，残念でした。

さて，誌面が少し余ったようなので，東京都の大山幸二さんからいただいた質問に回答したいと思います。

「1月号のぱーてぃハンズに書かれていたinkey$(0)だけのキーバッファクリアはどうやってするのでしょうか？」ということですね。

(で)氏が1月号で紹介したキーバッファクリアの方法をここで引用すると，

while(keysns( )):a$＝inkey$(0)：end while

となっています。1月号を持っていない方のために簡単に説明すると，keysns()というのはX-BASICの未公開命令で，キーバッファの中身を調べて空なら0，そうでなければ－1を返す命令です。inkey$(0)もkeysns( )同様，X-BASICの未公開命令です。これは，

a$＝inkey$(0)

のようにして，リアルタイムキー入力を実現する命令です。

しかし，この命令には，(で)氏が1月号で指摘しているように，キー入力がキーバッファに溜まってしまうという弱点があります。よく考えてみてください。inkey$(0)がキーバッファに溜まるということは，inkey$(0)はキーバッファの先頭から1文字取り出す命令だと解釈することができます。つまりキーバッファが空であるということは，

inkey$(0)＝""

の状態であるといえます。ゆえにinkey$(0)を使ったキーバッファクリアは，

while inkey$(0)＜＞"":endwhile

と書くことができます。わかっていただけましたか？

というところで，では，また来月。

リスト1 SHODO68K.BAS

```
10 float r
20 int x,y,hx,hy,k,yy,qa,qb,qc,qd
30 str f,fn
40 init()
50 while 1
60   fude()
70   keyin()
80 endwhile
90 end
100 /*
110 func fude()
120   msstat(qa,qb,qc,qd)
130   mspos(x,y)
140   if qc=-1 then r=r+0.5#
150   if qd=-1 then r=r-0.5#
160   if r<0 then r=0
170   cfill(x,y+yy,r,0)
180 endfunc
190 /*
200 func cfill(x,y,r,c)
210   int i,j
220   if r>=1 then {
230     mouse(2):circle(x,y,r-1,9,0,360,358)
240   } else mouse(1)
250   for j=0 to r-1
260     i=sqr(r*r-j*j)/1.4#
270     line(x-i,y-j,x+i,y-j,c)
280     line(x-i,y+j,x+i,y+j,c)
290   next
300 endfunc
310 func keyin()
320   k=asc(inkey$(0))
330   if k=14 and yy<512 and y>64 then {
340     for i=1 to 64
350       yy=yy+1
360       home(0,0,yy)
370       msarea(0,0,511,y-i)
380       msarea(0,0,511,511)
390     next
400     bar() }
410   if k=15 and yy>=64 and y<447 then {
420     for i=1 to 64
430       yy=yy-1
440       home(0,0,yy)
450       msarea(0,y+i,511,511)
460       msarea(0,0,511,511)
470     next
480     bar() }
490   if k=27 then command()
500 endfunc
510 func command()
520   cls
530   mouse(2)
540   r=0
550   locate 10,10:print"1. LOAD"
560   locate 10,11:print"2. SAVE"
570   locate 10,12:print"3. CLEAR"
580   locate 10,13:print"4. EXIT"
590   locate 10,15:print"0. END"
600   repeat
610     k=asc(inkey$)-48
620   until k>=0 and k<5
630   if k=0 then vpage(0):color 3:cls:end
640   if k=1 or k=2 then {
650     locate 27,18:print f
660     locate 10,18:input"Input File Name =",f
670     fn=f+".sho" }
680   if k=1 then apic_load(fn)
690   if k=2 then apic_save(fn,0,0,511,1023)
700   if k=3 then fill(0,0,511,1023,15)
710   mouse(1)
720   bar()
730 endfunc
740 func bar()
750   cls
760   for i=1 to yy/64:print".":next
770   for i=0 to 8:print"I":next
780   for i=1 to (512-yy)/64:print".":next
790 endfunc
800 func init()
810   screen 1,0,1,1
820   console 0,31,0
830   window(0,0,1023,1023)
840   vpage(1)
850   fill(0,0,511,1023,15)
860   mouse(4)
870   mouse(1)
880   msarea(0,0,511,511)
890   color[,51250]
900   color 5
910   bar()
920 endfunc
```

リスト2 WILD.C

```
/*****************************************************


        WILD   CARDER    Ver 1.01

                              平成4年6月14日

                       PROGRAMMED  By  野崎  哲也

******************************************************/
#include <stdio.h>
#include <doslib.h>
#include <stdlib.h>

struct FILBUF *buffer,buf;
unsigned char *file;
int atr;
unsigned char filename[256];
unsigned char command[1024];

/*****************************************************/
main(argc,argv)
int argc;
unsigned char *argv[];
{
        int dummy;
        int i=0;
        int num=0;

        buffer = 0xa0000;
        atr = 0x27;
        *filename = 0;
        printf("X68k Wild Carder Ver 1.01 Copyright 1992 By Tetsuya.Nozaki¥n");

        if ( argc <= 2) {
                printf("使用法:wild  ワイルドカードを使ったコマンド¥n¥n");
                exit(-1);
        }

        for (i=2;i<argc;i++) {
                switch (*argv[i]) {
                        case '/':
                        case '-':
                                break;
                        default :
                                num=i;
                }
        }
        if ( num == 0 ) {
```

```
                        printf("ファイルの指定がされていません。¥n
");

                        exit(-1);
        }
        strcpy(filename,argv[num]);

        dummy=FILES(buffer,filename,atr);

        if (dummy != 0) {
                plus(filename);
                dummy=FILES(buffer,filename,atr);

                if (dummy != 0) {
                        printf("該当するファイルが見つかりません。¥
n");

                        exit(-1);
                }
        }

        while (dummy == 0) {
                cut(filename);
                strcat(filename,buffer->name);

                *command=0;
                strcpy(command,argv[1]);

                for (i=2;i<argc;i++) {
                        strcat(command," ");
                        if (i == num) strcat(command,fil
ename);
                                        else    strcat(command,a
rgv[i]);
                }
                printf("¥n¥x1b[42m『%s』で実行します。¥x1b[33m
¥n",filename);
```

```
                        system(command);

                        dummy = NFILES(buffer);

                }
}

/****************************************************/
cut(po)
unsigned char *po;
{
                unsigned char *poi=po;

                while (*(++poi));

                while ( (   (*(--poi)!='¥¥')  && (*poi != ':')   )
&& (poi>=po) );

                *(++poi)=0;
}


/****************************************************/
plus(po)
unsigned char *po;
{
                unsigned char *poi=po;

                while (*(++poi));

                poi--;
                if (   ( (*poi) != '¥¥' )   &&   ( (*poi) != ':'
)   ) strcat(po,"¥¥");
                strcat(po,"*.*");
}
```

(影)のぱーてぃハンズ

vdisp関数のライブラリ制作

前回掲載したvdisp.fncは打ち込んでもらえたでしょうか。あんまり使いもんにならないかもしれませんけど。今月のぱーてぃハンズではvdisp.fncを例にとって，X-BASICの外部関数からC言語のライブラリを作成する方法を解説したいと思います。

ライブラリ作成のお約束

ライブラリを作成するには，いくつかの決まりごとがあります。以下にそれらをまとめておきました。

1）アセンブラプログラムで破壊してもいいレジスタはd0/d1/d2，a0/a1/a2です。そのほかのレジスタをアセンブラプログラム内で使用する場合は，プログラムの入口でレジスタの値を保存しておき，出口で値を復帰させるようにします。

2）外部関数名の頭に“_”を付けたラベル名をプログラムの先頭に必ず付けます。

つまり先月掲載したvdisp.sの9〜45行をバッサリと削除して，48のラベルvdispを_vdispに変更します。これで作業の80%は終了しています。残り20%はパラメータの取り込みと，レジスタの保護です。幸いvdisp.sは壊してもいいレジスタしか使用していないので，レジスタの保護をする必要はありません。すると残りはパラメータの取り込みということになりますが，Cの場合は外部関数と違いスタックフレームに関数の型などの情報は一切なく，純粋に引数のみが格納されています。プログラムの入口では現在のspに戻りアドレスが格納されていますので，引数は4(sp)から始まります。また引数はintなので，スタックフレームは4バイト使用します。したがって，49行は，

　tst.l　4(sp)

ですみます。これだけの変更でライブラリを作成することができます。vdisp_lib.sを入力したらアセンブルしてください。リンクする必要はありません。

次にBASIC.DEFに，

　I vdisp(I)

を加え，BASIC.Hに，

　void vdisp(int);

を追加します。あとはvdisp.fncを使ったプログラムをコンパイルするときに，

　A>cc /W game.bas vdisp_lib.o

とすれば，無事コンパイルできると思います。

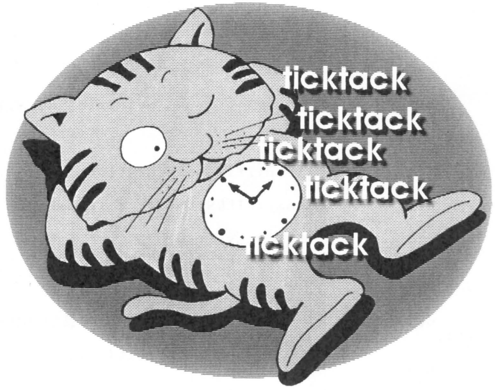
リスト

```
*
*  C言語のライブラリ用
*
GPIP:           equ     $e88001

        .include        iocscall.mac

        .text
_vdisp:
        tst.l   4(sp)
        bne     vdisp_rts
vdisp2:
        lea.l   GPIP,a1
        IOCS    _B_BPEEK
        btst.l  #4,d0
        beq     vdisp2  * 帰線なら
vdisp3:
        lea.l   GPIP,a1
        IOCS    _B_BPEEK
        btst.l  #4,d0
        bne     vdisp3  * 表示なら
        clr.l   d0
vdisp_rts:
        rts

vdisp4:
        cmpi.l  #1,4(sp)
        bne     vdisp_rts

        lea.l   GPIP,a1
        IOCS    _B_BPEEK
        btst.l  #4,d0
        bne     vdisp4  * 表示なら
vdisp5:
        lea.l   GPIP,a1
        IOCS    _B_BPEEK
        btst.l  #4,d0
        beq     vdisp5  * 帰線なら
        clr.l   d0
        rts

        .end
```

吾輩はX68000である
[第16回]

時を刻む

Izumi Daisuke
泉　大介



時間は誰の上にも平等に流れ
吾輩のなかにも刻み込まれていく
眠っていても起きていても

前回はデバッガでフロッピーディスクを吐き出すというテーマで，メモリマップトI/Oについてお届けした。ついでにD-フリップフロップの原理について説明し，吾輩がどのようにして「特定のビットが1になっているときに有効なデータ」を扱っているのかを紹介した。

かつては腕白な電気工作小僧がパーツ屋にたむろし，部品を買ってきてはお風呂の水位検知機やラブラブチェッカー，鉱石ラジオから4石スーパーラジオ，さらにはオーディオアンプや無線機などを自作していたものだが，最近では右を向いても左を向いても「買ってくればいい」族が氾濫していて寂しいかぎりである。とある腕白小僧は，自宅のコンセントから取り出したAC100Vをトランスでコンバートし，その先に直流モータを繋ぐという無謀な実験を敢行した。直流モータは火花を散らしながら転げ回り，この小僧とその友人をおおいに楽しませたものである。哀れな直流モータの命が数秒で文字どおり燃え尽きたことはいうまでもない。

吾輩がこうしてこの世に存在しているのも，成人した腕白小僧があればこそである。D-フリップフロップが，諸兄の関心の向きを多少なりとも刺激できればいいのだが。

さて，今回は，吾輩が起きているときも寝ているときもひたすら動き続けている内蔵時計に焦点を当ててみたい。

◆時の流れに身を……

パソコンが時計機能をもっているのは当然のように思われているが，ポケコンなどではむしろ時計機能をもっていないのがふつうである。御仁はカシオのポケットコンピュータからコンピュータの世界に入ったらしいのだが，これを目覚し時計代わりに使うためには，

```
FOR I=0 TO 1000 : NEXT
```

のような空のループを作成して1秒分の時間待ちを作り，さらにこれを8×60×60回繰り返すループを作成して8時間の時間待ちを作るという，涙ぐましい努力が必要なのであった。空のループを何回実行すれば1秒になるのかというのがいちばんの問題で，時計とにらめっこをしながら，30秒，1分，5分とチェックをしていったということだ。それでも朝までに数分ずれてしまうことは珍しくない。さらに，一晩中ポケコンを動かしているために電池は数日ともたず，目が覚めてポケコンを見てみると，液晶が消えてしまっていたこともあったという。新しい電池を買いにいくたびに，御仁はしみじみと「時は金なり」という言葉を実感したことだろう。

我々パソコンが時計を内蔵しているのは，主としてファイルにタイムスタンプを押すためである。同じファイル名のファイルがA，B 2つのディスクに入っている場合，どちらのファイルがより新しいファイルなのか諸兄が判断できるように，というわけだ。ポケコンは基本的に内蔵しているRAMディスクにファイルを保存しておくだけなので，同じファイル名で作成日付の異なる2つのファイルが存在する余地はない。この違いが最も大きな理由だろう。

ポケコンにもの足りなくなった御仁が購入したMZ-2000君に付属のBASICには，TI$というシステム変数が用意されていた。この変数はMZ-2000君に内蔵された時計が刻んでいる時刻を表しており，たとえば午前9時30分20秒は“093020”と表現される。もちろん，TI$は1秒ごとに更新されるようになっている。ただし，時計が用意されているからといって，MZ-2000君がファイルにタイムスタンプを押していたというわけではない。時計が動いているのはMZ-2000君が通電されている間だけであり，起動時には再び“000000”からスタートする。さらには吾輩のように日付を管理する機能もなかったからである。それでも1秒を測る空ループを作成する必要がないということは，御仁にとっては大きな魅力だったことだろう。

```
10 IF TI$<>"074500" GOTO 10
20 PLAY "CDE" : GOTO 20
```

とプログラムしておくだけで，朝までぐっすり眠ることができるのだから。このプログラムはGOTOスパゲティ文化の香りが存分に溢れていて，また別の意味で興味深いところである。

◆吾輩の時計とその機能

吾輩の時計にはRP5C15というICが使用されている。このICは年月日・曜日・時刻を管理することができ，決められた時刻になるとアラームを発する機能を備えているのが特長である。吾輩は本体前面の電源スイッチが切られている間もこのタイマに電源を供給し続けるようになっており，仮に本体後面の電源スイッチまで切られてしまっても，さらに内蔵電池で時計を駆動し続けるように作られている。現在のパソコンで時計がいかに大切にされているかを示す例といえるだろう。

決められた時刻になって時計がアラームを発すると，吾輩は専用ディスプレイテレビをつけたり，あらかじめ指定されていたプログラムを実行することができる。そんなわけで吾輩が御仁のもとにやってきて以来，御仁の就寝前の2行のプログラミング習慣はすっかりなりをひそめてしまった。

内蔵時計は，秒をカウントするもの，分をカウントするもの，時をカウントするもの，といったいくつかのレジスタによって構成されている。これらのレジスタは4ビットのデータを保持することができ，例によって例のごとく吾輩のメモリにマップされている(図1)。図中斜線を入れてあるのは，レジスタにマップされていない無効なビットである。これらの無効ビットはすべて1になるようになっているが，デバッガのDコマンドではダンプ表示することができないのでご注意いただきたい。内蔵時計のレジスタを，デバッガのMEコマンドを使って表示してみたのが図2である。4ビットは1桁の16進数と同じなので，表示されていくデータの最下桁だけを集めていけば，右側に示したように年月日と曜日，そして現在時刻を抽出することができる。

◆ABCD……？

図2を見てお気づきかと思うが，内蔵時計のレジスタは少々面白い形式でデータを保持するようになっている。通常4ビットあれば，0000_B～1111_B，つまり0～15までの数を表現できるのだが，これらのレジスタは，

E8A001$_H$：1秒の位

E8A003$_H$：10秒の位

という形式で秒を表現しているのである。現在の秒数が19秒，つまり，

E8A001$_H$：F9$_H$

E8A003$_H$：F1$_H$

だったとすると，次は，

E8A001$_H$：FA$_H$

E8A003$_H$：F1$_H$

とはならずに，

E8A001$_H$：F0$_H$

E8A003$_H$：F2$_H$

図1　内蔵時計のレジスタ

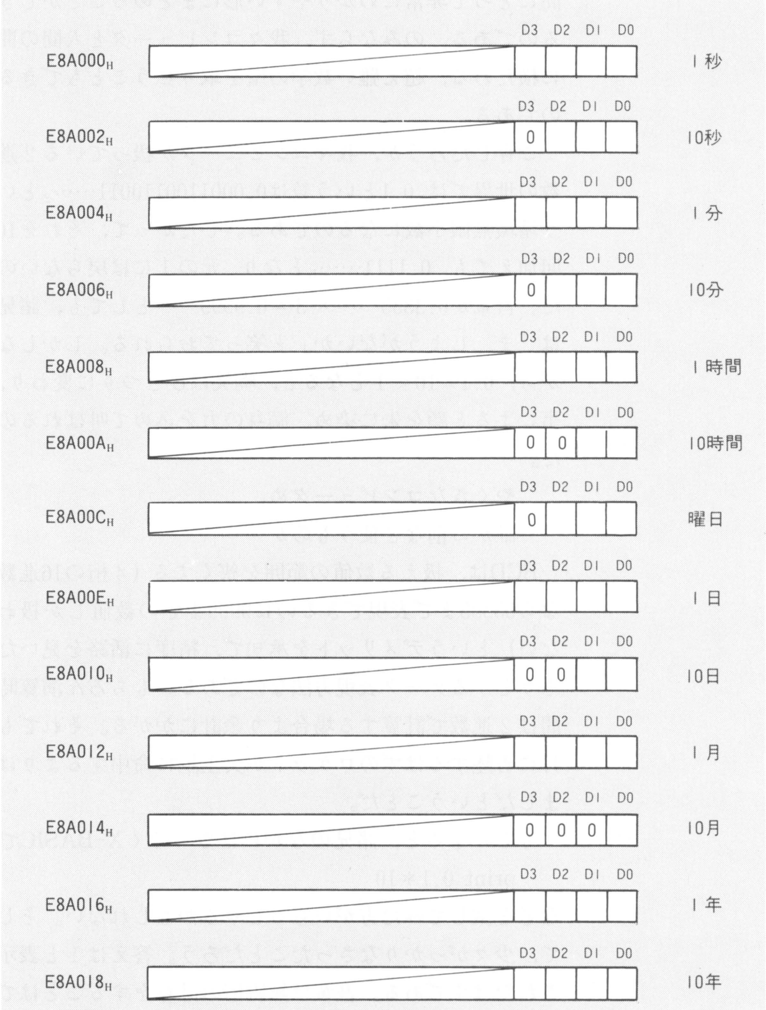


図2　内蔵時計のレジスタを表示してみる

```
-me e8a000
00E8A000    FFF2 : }52秒
00E8A002    FFF5 :
00E8A004    FFF7 : }7分
00E8A006    FFF0 :
00E8A008    FFF3 : }3時
00E8A00A    FFF0 :
00E8A00C    FFF2 : 火曜日
00E8A00E    FFF8 : }18日
00E8A010    FFF1 :
00E8A012    FFF8 : }8月
00E8A014    FFF0 :
00E8A016    FFF2 : }12年
00E8A018    FFF1 :
00E8A01A    FFF8 :
```

と20秒を示すようになる。せっかく4ビットのデータを使っていながら，1010_B～1111_Bの6つのデータは使われないままお払い箱である。

このように2進数を使って10進数を表現する方法は，BCD（Binary Coded Decimal：2進化10進）と呼ばれている。この方法を使えば，999＋1＝1000は0999_H＋0001_H＝1000_Hと表現できることになり，2進数，ひいては16進数の世界で生きているコンピュータが扱う数を，人間にとって非常にわかりやすい形にまとめることができるのである。のみならず，我々コンピュータと人間の間に横たわる，超え難い数学の壁を取り払うこともできるのである。

ご存じだろうか，我々コンピュータが扱っている2進数の世界では，0.1という数は$0.0001100110011\cdots\cdots_B$という循環無限小数になるのである。したがって，それを10回加えても，$0.1111\cdots\cdots_B$となり，元の1には戻らないのだ。吾輩が0.3333……×3＝0.9999……としても，諸兄は「ま，しようがないか」と笑っておられる。しかしながら，0.1×10＜1となると，嘲笑はひきつりに変わり，事によると顔を朱に染め，満身の力を込めて叫ばれるのだ。

　やくざなコンピュータめ，
　誰がお前など使うものか

BCDは，扱える数値の範囲を狭くする（4桁の16進数なら65535まで表現できるのに9999までの数値しか扱わない）というデメリットを承知で，精度に活路を見いだそうとするデータ表現方法なのである。もちろん演算時間は2進数で計算する場合より余計にかかる。それでも，月に着陸するはずのロケットが冥王星に命中するよりはましだということだ。

もしかすると，諸兄のなかにはさっそくX-BASICで，

```
print 0.1*10
```

などと試してみた方がいらっしゃるかもしれない。そして，少々がっかりなさったことだろう。答えは1と表示されたはずである。吾輩は無限桁の計算をすることはできないため，どこか適当なところで四捨五入せざるを得ない。コンピュータ風にいうなら，ゼロ捨1入である。この下のほうの桁でのちょろまかしが，結果的には正しい計算結果をもたらしているのである。それでももちろん限界はある。簡単なところでは，

```
print 0.1+0.1+0.1+……+0.1=1.0
```

というのがそうである。0.1を生真面目に10回足し，その結果が1.0となるかどうかを確かめようというのだ。最後の「＝1.0」がなければ，結果は1と表示される。はたして本当に計算結果も1なのか。それを確かめていただきたい。もし両辺が等しければ結果は－1となる。これはX-BASICでは「真」という意味だ。異なっていれば答えは0，すなわち「偽」である。吾輩の数値計算の限界を探る旅も，また面白いものだ。ぜひ，試してみていただきたい。

もちろん内蔵時計にはこんな精度は必要ない。にもかかわらずBCDが採用されているのは，このICの出生と関係がある。

吾輩の内蔵時計は，本来デジタル時計用のものなのである。デジタル時計は7セグメントのLED（7本のバーで8の字を表現している例のあれ）を使って数字を表示する。そして世の中には4ビットで0～9のデータを与えると，7セグメントのLEDの該当するバーを光らせるのに必要な7ビットのデータに変換してくれるICが存在するのだ。汎用の時計用ICなれば，アラーム機能が付いているのもまた道理というわけである。

ちなみにこの節の見出しに掲げたABCDは，Add BCDというれっきとしたMC68000の命令である。

```
abcd Dn,Dn
abcd -(An),-(An)
```

という形式で，データレジスタどうしのBCD加算，あるいは，メモリに格納されたデータどうしのBCD加算がサポートされている。ついでに実験してみていただきたい。

◆もう36時間も連続で働いています

内蔵時計に関連して，吾輩のSRAMに書き込まれている面白いデータについて紹介しておこう。ちょいとデバッガを使って，アドレス$ED0040_H$のデータを表示してみていただきたい。ここに書き込まれているロングワードデータは，吾輩が今日までに合計何時間働いてきたかを表すデータである。もしこれまでにSRAMを再初期化するような事態に陥っていなければ，吾輩が諸兄のもとでどれだけの献身をしてきたかが読み取れるはずである。ロングワードデータの単位は秒。書き込まれているデータを日，月，あるいは年に変換して，吾輩が諸兄の生活の何%を占めているのかを計算してみるのも面白かろう。

　この5年間，
　自分の人生の20%はX68000とともにあった

そこまで愛していただければ吾輩としては本望である。

ついでにもうひとつ。電源を入れられてから吾輩が何時間連続で奉仕しているかを保持しているワークエリアがある。これはアドレスが公開されていない内部ワークなのだが，IOCSコール$7F_H$を使えば簡単にデータを得ることができる。引数はない。単に，

```
moveq #$7f,d0
trap #15
```

とすれば，D0.lに経過時間（単位は1/100秒）が，D1.lに経過日数（単位は日）が返される。

＊　＊　＊

目前に旅行を控えて御仁の仕事は現在頂点に達している。吾輩も少々過労気味だ。今回はこのあたりでお開きということにしよう。

整数演算のアルゴリズム

Murata Toshiyuki 村田　敏幸

今回と次回の2回にわたって整数演算について話をしていきます。数学は苦手な人も，1つひとつ村田氏がていねいに解説しているのできっと理解できることでしょう。まず今回は，整数演算の基礎として乗算/除算とその応用，そして平方根の求め方です。

Assembler

2回に分けて算数する。今月は整数演算の話をしてみたい。とりあえず，いくつかの演算アルゴリズムを知っておいてもらおうと思う。

乗算

まずは基本ということで，乗算から始めよう。68000には乗算命令mulsとmuluが用意されているわけだが，16ビット数どうしの積しか求められないため，扱う数がもっと大きくなると自前の乗算ルーチンを用意する必要が出てくる。そんなとき，真っ先に思いつくのは乗算を加算に置き換える方法だろう。m×nなら，ループを回してmをn回足し合わせればいい。ただ，この方法はnが大きくなると非常に時間がかかる。32ビット×32ビットの場合，最悪43億回近くもの加算を繰り返す羽目になるわけだ[1)]。

そこで，より実用的な方法を紹介しよう。このアルゴリズムは乗算命令をもたないプロセッサで乗算を実現するのによく使われる方法で，早い話が2進数レベルで積を筆算で求めるものだ。

図1に2進4桁の数どうしを筆算で掛ける様子を示した。10進の場合と同様に，

1)　乗数を下位から1桁ずつ取り出して被乗数との積を個別に求める。

2)　桁位置を揃えて足す。

という手順を踏んでいるのがわかると思う。1)のステップが10進の場合よりも簡単になっていることに注目してほしい。積とはいっても2進数1桁倍だから，0倍と1倍しかないのだ。また，2)のステップでの桁位置を揃える操作は，ビットシフトで実現できることにも気づくだろう。結局，この筆算の手順をそのままプログラムにすれば，乗数のビット数に比例した回数の加算とビットシフトだけで構成された乗算ルーチンが作れる。

ところで，何進数だろうとm桁×n桁の積は最大でm＋n桁の数になる。2進でいうと，mビット×nビットの積はm＋nビットの数になりうるということだ。ただ，プログラムのうえでは，被乗数，乗数，積を格納する領域の大きさをわざわざ変えることはあまりない。通常は演算過程で生まれる最大値が格納できるだけの記憶域をあらかじめ見積もり，全変数をその大きさに揃えるものだ。したがって，乗算ルーチンもnビット×nビットをnビットで求めるものを用意しておけばたいてい間に合う。68000の場合，ふつうに扱えるデータの最大長は32ビットだから，32ビット×32ビットを32ビットで求めるルーチンがあれば実用上は十分だ。

というわけで，リスト1に32ビット×32ビット＝32ビットの乗算サブルーチン例を示す。動作試験用のプログラムはとくに用意しないが，適当な数どうしを掛ける小さなプログラムを書いて，電卓か，ほかのプログラム実行結果と照合してみてほしい。オーバーフローの判定はしておらず，もし，フロったとしても何食わぬ顔で結果の下位32ビットだけを返す。引数はd0.l, d1.lに入れて渡し，結果はd0.lに返

1)　ちなみに，65536回のdbraループを2重にして約43億回の空ループを組み，クロック周波数10MHzの68000で実行すると1時間以上かかる。

図1　2進法の乗算

```
        1110
 ×)     1101
 ------------
        1110
      1110
     1110
 ------------
    10110110
```

リスト1　MUL32.S

```
*       d0×d1=d0

        .xdef   muls32
        .xdef   mulu32
*
        .text
        .even
*
muls32:
mulu32:
        movem.l d1-d3,-(sp)

        move.l  d0,d2           *d2 = 被乗数
        moveq.l #0,d0           *d0 = 積格納先
        moveq.l #32-1,d3
loop:   add.l   d0,d0           *中間結果を左シフト
        add.l   d1,d1           *乗数の上位1ビットが
        bcc     next            *  0なら何もしない
        add.l   d2,d0           *  1なら被乗数を足し込む
next:   dbra    d3,loop         *32ビット分繰り返す

        movem.l (sp)+,d1-d3
        rts

        .end
```

る。サブルーチンmuls32が符号付き数用，mulu32が無符号数用だ……が，両者の実体は同じもの。実は，負の数を２の補数表現で表す世界でｎビット×ｎビットの下位ｎビットを求めるだけなら，符号を考慮する必要がない。

再び図１を見てもらいたい。被乗数と乗数を４ビットの符号付き数とみなすと，その値は10進の－２と－３だ。結果の下位４ビットを符号付き４ビット数とみなして10進に直すと６，ちゃんと，－２と－３の積になっている。値をいろいろ変えてみてもこの関係は常に成り立つ。念のため，正×負や負×正の場合でも確認しておいてほしいと思う。

では，リスト１の流れを見ていこう。基本的には図１と同様の操作を行っているのだが，微妙に変形してある。中間結果は最後にまとめて足すのではなく，求めたその場で最終結果格納用のレジスタにどんどん足し込むようになっており，また，中間結果を求める順序も通常の筆算のように乗数の下位桁側から処理していく代わりに，上位桁側から処理するように変えてある。ともにプログラムの構造を簡潔にするための小細工だ。まず被乗数d0をd2に移動しておいて最終結果格納用にd0を空けて（13～14行）から，16行以降のループに入る。ループ頭の16行はちょっと置いて，17～19行を見てほしい。加算命令２つと条件分岐１つで“乗数を上位から１ビット取り出し，被乗数と掛けて，d0に足し込む”という処理を実現している。すでに触れたように，被乗数×１ビットの乗算は０倍か１倍かのどちらかしかなく，０倍を足すのは何もしないのと同じことなので，“乗数から取り出した１ビットが１だったら被乗数d1をd0に足す”と読み変えて，単純な条件判断と加算に帰着させている。乗数d1の上位ビットを取り出して０か１かを判定するのに，17行の，

```
add.l	d1,d1
```

によって変化するccrのＣビットを使っていることがわかれば，何をやっているかは明らかだろう。この17行は，

```
lsl.l	#1,d1
```

と書いても同じ意味であり，d1を１ビット左シフトして最上位ビットをccrのＣビットに追い出している[2]。以下，ループが回るごとにd1の最上位ビットから１ビットずつが，順にＣビットに追い出され，直後のbccで処理が振り分けられるというわけだ。

こうしてループがひと回りして16行に戻った時点を考えよう。d0には被乗数×乗数の第31ビット（最上位ビット）の値が入っている。２周目では被乗数×乗数の第30ビットをd0に足し込むことになるが，単純に足すわけにはいかず，桁の重みを考慮する必要

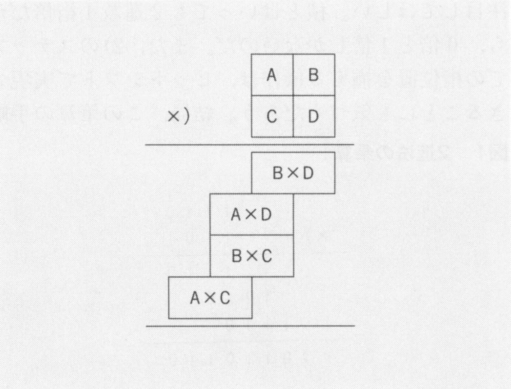
リスト2　MUL32.S（修正版）

```
 1: *        d0×d1=d0
 2:
 3:          .xdef    muls32
 4:          .xdef    mulu32
 5: *
 6:          .text
 7:          .even
 8: *
 9: muls32:
10: mulu32:
11:          movem.l d1-d3,-(sp)
12:
13:          move.l   d0,d2             *d2 = 被乗数
14:
15:          swap.w   d1                *被乗数下位16ビット
16:          moveq.l #16-1,d3           *  ×乗数上位16ビット
17: loop1:   add.w    d0,d0             *←
18:          add.w    d1,d1             *←
19:          bcc      next1             *
20:          add.w    d2,d0             *←
21: next1:   dbra     d3,loop1          *
22:
23:          swap.w   d1                *+被乗数32ビット
24:          moveq.l #16-1,d3           *  ×乗数下位16ビット
25: loop2:   add.l    d0,d0             *
26:          add.w    d1,d1             *←
27:          bcc      next2             *
28:          add.l    d2,d0             *
29: next2:   dbra     d3,loop2          *
30:
31:          movem.l (sp)+,d1-d3
32:          rts
33:
34:          .end
```

リスト3　MUL32.S（最終版）

```
 1: *        d0×d1=d0
 2:
 3:          .xdef    muls32
 4:          .xdef    mulu32
 5: *
 6:          .text
 7:          .even
 8: *
 9: muls32:
10: mulu32:
11:          movem.l d1-d3,-(sp)
12:
13:          cmp.l    d1,d0             *d0≧d1を保証する
14:          bcc      skip              *
15:          exg.l    d0,d1             *
16: skip:                               *d0.l = AB
17:                                     *d1.l = CD
18:          swap.w   d0
19:          move.w   d0,d2             *d2.w = A
20:          beq      mul0              *d1.l ≦ d0.l < 65536
21:          swap.w   d0
22:
23:          swap.w   d1
24:          tst.w    d1
25:          beq      mul1              *d1.l < 65536 ≦ d0.l
26:
27:                                     *d1.w = C
28:          move.w   d0,d3             *d3.w = B
29:          mulu.w   d1,d3             *d3.l = B×C
30:          swap.w   d1                *d1.w = D
31:          mulu.w   d1,d0             *d0.l = B×D
32:          mulu.w   d1,d2             *d2.l = A×D
33:
34:          swap.w   d0                *d0.w = B×Dの上位.w
35:          add.w    d2,d0             *d0.w += A×Dの下位.w
36:          add.w    d3,d0             *d0.w += B×Cの下位.w
37:          swap.w   d0                *d0.l = AB×CD
38:
39: retn:    movem.l (sp)+,d1-d3
40:          rts
41: *
42: mul0:    swap.w   d0                *d1.l ≦ d0.l < 65536
43:          mulu.w   d1,d0             *d0.l = 0B×0D
44:          bra      retn
45:
46: mul1:    swap.w   d1                *d1.l < 65536 ≦ d0.l
47:          mulu.w   d1,d0             *d0.l = B×D
48:          mulu.w   d1,d2             *d2.l = A×D
49:          swap.w   d0                *d0.w = B×Dの上位.w
50:          add.w    d2,d0             *d0.w += A×Dの下位.w
51:          swap.w   d0                *d0.l = AB×0D
52:          bra      retn
53:
54:          .end
```

図2　65536進2桁の乗算



がある。このつじつまを合わせているのが先ほど飛ばした16行だ。前回のループまでの中間結果の和を2倍することにより桁の重みを反映している。筆算で桁をずらして書くのと同じ感覚だ。

さて，効率の点ではリスト1は最良とはいえず，けっこう改良の余地がある。小さなところでは14行で最終結果格納用のd0を0クリアしている部分。d0は計32回左シフトするので，クリアせずにゴミを入れたままにしておいても結果には影響しない。また，ループの前半16周が終了した時点でのd0の値のうち，上位16ビットは，続くループで16回左シフトするうちに，やはり最終結果32ビットの圏外に消えてしまう。このことに着目すると，前半16周はワード演算ですむことがわかる。その線でリスト1を書き直すと，リスト2のようになった。

ここで，リスト2の最初のループの意味をもう少し掘り下げてみよう。このループを抜けた時点でd0に入っている値は，被乗数の下位16ビットと乗数の上位16ビットとの積（の下位16ビット）だ。16ビット×16ビットの乗算なのだから，このループはmulu命令ひとつに置き換えてしまうことができる。これを踏まえて2番目のループのほうも見てみると，こちらで計算しているのは被乗数32ビット×乗数の下位16ビットだ。便宜上，被乗数の上位下位16ビットをそれぞれAとBに分割し，乗数のほうも同様にCとDに分割して考えると，1番目のループでは，

B×C

を，2番目のループでは，

AB×D

を求めているわけだ。で，AB×Dは，

A×D

を16ビットシフトしたものと，

B×D

を足したものに等しい。つまり，2番目のループもmulu命令2つに置き換えられる。最終的に，32ビット×32ビット＝32ビットの乗算は，mulu命令3つの結果を16ビットのシフトを交えて足し合わせることで実現できるのだ。これは，32ビット数を 2^{16}＝65536進2桁の数とみなすとわかりやすい(図2)。

乗算ルーチンの最終版をリスト3に示す。32ビット×32ビットの乗算ルーチンは，むしろ，

16ビット×16ビット
32ビット×16ビット
16ビット×32ビット

の用途に使われることが多いことを考慮し，リスト3では適当に場合分けをして実用上の効率を稼ぐようにしてある。被乗数と乗数のどちらか，あるいは，両方が16ビットに収まる場合を特別扱いして処理を振り分けているのだ。

2) 10進数を10倍すると1桁左にずれて最下位桁に0が入るが，これと同様に，2進数は2倍すると1桁左にずれる(左にビットシフトする)。逆に，2進数をn桁左シフトすると値は 2^n 倍になり，右シフトすれば値は 2^n 分の1になる。これも10進数を左右にn桁ずらすと値が 10^n 倍になったり，10^n 分の1になったりするのに対応する。

階乗と累乗

乗算ルーチンを応用して，階乗と累乗を求めるサブルーチンを作ってみた(リスト4，5)。どちらも単純なプログラムで，階乗ルーチンのほうは1からd0までの値をループで掛けるだけ，d0のd1乗を求める累乗ルーチンもループでd0をd1－1回掛けるだけという代物だ。すかさず，効率を上げる方向に話をもっていこう。

まず，階乗だ。nの階乗n！は，nが少し増えただけでとんでもなく大きな数になる。100！にもなると実に10進158桁という数だ。13！ですら6,227,020,800になり，32ビット整数では収まり切らない。結局，リスト4のサブルーチンfactの有効な引数の範囲はわずか0～12に制限される。引数が13通りしかありえないのなら，結果もまた13通りだ。そのくらいなら，毎回馬鹿正直に乗算を行わなくても，あらかじめ0！～12！を計算してテーブルにしておくという手段が使える。リスト6はこの姑息な方法で作り直した階乗の計算（？）ルーチンだ。実際の階乗計算はアセンブラにやらせている。

累乗のほうは，

リスト4　FACT32.S

```
*       階乗

        .xdef   fact32
        .xref   mulu32
*
        .text
        .even
*
fact32:
        movem.l d1-d3,-(sp)

        move.l  d0,d2               *d2 = n
        moveq.l #1,d0               *d0 = 1!,2!,3!,...
        moveq.l #1,d1               *d1 = 1, 2, 3, ...

        subq.l  #1,d2               *dbraを考慮
        bcc     loop1
        bra     retn                *0! = 1

        swap.w  d2                  *1からnまでを
loop0:  swap.w  d2                  * 掛け合わせる
loop1:  jsr     mulu32              *
        addq.l  #1,d1               *
        dbra    d2,loop1            *
        swap.w  d2                  *
        dbra    d2,loop0            *

retn:   movem.l (sp)+,d1-d3
        rts

        .end
```

リスト5　POWER32.S

```
*       累乗

        .xdef   power32
        .xref   mulu32
*
        .text
        .even
*
power32:
        movem.l d1-d2,-(sp)

        move.l  d1,d2               *d2 = 指数
        move.l  d0,d1               *d1 = 仮数 x
        moveq.l #1,d0               *d0 = x^0 = 1

        subq.l  #1,d2               *dbraを考慮
        bcc     loop1
        bra     retn                *x^0 = 1

        swap.w  d2                  *xをn回掛け合わせる
loop0:  swap.w  d2                  *
loop1:  jsr     mulu32              *
        dbra    d2,loop1            *
        swap.w  d2                  *
        dbra    d2,loop0            *

retn:   movem.l (sp)+,d1-d2
        rts

        .end
```

$$y = a + b$$

のとき，

$$x^y = x^a \times x^b$$

という初歩的な数学の知識を使うと，とくに指数が大きなときの実行速度を飛躍的に向上させることができる。

この考え方では，指数yを2^nの和の形で表す。たとえば，y＝13なら，

$$y = 1 + 4 + 8$$

のように分解する。すると，

$$x^{13} = x^1 \times x^4 \times x^8$$

だ。ここで，

$$x^2 = x \times x$$
$$x^4 = x^2 \times x^2$$
$$x^8 = x^4 \times x^4$$
$$\vdots$$

だから，xを次々に2乗していくと，x^2，x^4，x^8，……が順次求まる。こうして求めたx，x^4，x^8を掛け合わせれば，x^{13}が得られるわけだ。

乗算の回数を数えてみると，x^4を求めるまでに2回，そこからx^8を求めるためにもう1回，最後にx，x^4，x^8を掛け合わせるために2回の計5回となり，正直に12回掛けるよりも大分少ないことがわかる。指数がもっと巨大だと効果も大きい。指数が32ビットの場合，その値が億単位であろうとも，最悪62回の乗算ですむのだ。

リスト7にこの考え方の実現例を示す。コンピュータ内部の数値はすでに2進表現になっているから，わざわざ指数を2^nの和に分解する必要がないことに注目しよう。たとえば，13は2進数で表すと，

$$1101_B$$

になり，立っているビットを拾うと，

$$0001_B = 1$$
$$0100_B = 4$$
$$1000_B = 8$$

より，

$$13 = 1 + 4 + 8$$

だということがすぐわかる。ループの中でx，x^2，x^4，x^8，……を順次求め，指数を右に1ビットシフトしたときのCビットが1か0かで，いま求めたx^nを使うかどうか決めればよいということだ。

なお，リスト7の乗算の回数は上の説明よりも常に2回多い。初期値を1にしてこれにx^nを掛けていくという構造のために1回，ループを抜ける直前に不要なx^nを計算しているのでもう1回だ。多少工夫すれば，この無駄な乗算を省くことはできるので検討してみてもらいたい。

除算

除算は乗算以上に68000が不得手とする演算といえる。68000の除算命令divs，divuは32ビット÷16ビットの商と余りをそれぞれ16ビットで求めるだけの能力しかない。しかも，32ビット÷16ビットの商は16ビットで収まらずにオーバーフローする可能性がある。被除数があまり大きくないか，除数が十分大きいかのどちらかの場合でないと安心して使えない。ある程度の大きさの数を扱うプログラムではやはり自前の除算ルーチンが必須だろう。

例によって安直にやるなら，被除数から除数を引いていって何回引けたかを数えれば除算が実現できる。ただし，40億÷1などという例を挙げるまでもなく，この方法は場合によっては死ぬほど遅い。正しい対応は乗算のときと同様に2進法の筆算をソフトウェアでまねることだ。

図3に5ビット数÷3ビット数を筆算で求める様子を示す。手順自体は10進数のときとなんら変わらない。ただ，またまた2進法の利点で1桁ずつ商を

リスト6　FACT32.S(最終版)

```
*       階乗

        .xdef   fact32
*
        .text
        .even
*
fact32:
        cmpi.l  #13,d0              *n≧12なら
        bcc     flow                *  エラー

        lsl.w   #2,d0               *テーブルから
                                    *  n!を引いてくる
        move.l  facttbl(pc,d0.w),d0
        rts

flow:   moveq.l #-1,d0
        rts
*
facttbl:                            *0!～12!の表
        .dc.l   1
        .dc.l   1
        .dc.l   1*2
        .dc.l   1*2*3
        .dc.l   1*2*3*4
        .dc.l   1*2*3*4*5
        .dc.l   1*2*3*4*5*6
        .dc.l   1*2*3*4*5*6*7
        .dc.l   1*2*3*4*5*6*7*8
        .dc.l   1*2*3*4*5*6*7*8*9
        .dc.l   1*2*3*4*5*6*7*8*9*10
        .dc.l   1*2*3*4*5*6*7*8*9*10*11
        .dc.l   1*2*3*4*5*6*7*8*9*10*11*12

        .end
```

リスト7　POWER32.S(最終版)

```
*       累乗

        .xdef   power32
        .xref   mulu32
*
        .text
        .even
*
power32:
        movem.l d1-d3,-(sp)

        move.l  d1,d3               *d3 = 指数
        move.l  d0,d1               *d1 = x
        moveq.l #1,d0               *d0 = 1
        bra     next

loop:   lsr.l   #1,d3               *指数の最下位ビットが
        bcc     skip                *  1ならば
        jsr     mulu32              *  x^1,x^2,x^4,x^8,...を
                                    *  d0に掛ける

skip:   move.l  d0,d2               *x^1,x^2,x^4,x^8,...を
        move.l  d1,d0               *  d1に求める
        jsr     mulu32              *
        move.l  d0,d1               *
        move.l  d2,d0               *

next:   tst.l   d3                  *指数が0になるまで
        bne     loop                *  繰り返す

retn:   movem.l (sp)+,d1-d3
        rts

        .end
```

立てる部分は単純になっている。10進の場合は0〜9の数のうちどれが立つかを，ときには試行錯誤して調べるしかないわけだが，2進の場合は商の各桁は0か1にしかならない。どちらになるかは単に大小関係を比べるだけですぐわかる。勘を働かせる能力のないコンピュータでも機械的に実現できるのだ。

勘といえば，人間が除算を筆算で行うときには，最初に被除数と除数の桁を合わせる操作をほぼ無意識に行う。図3の場合，除数が3桁だから，

$1_B \div 100_B$

$11_B \div 100_B$

は考えず，最初から被除数の上位3桁に注目して，

$110_B \div 100_B$

に取り掛かるだろう。プログラムではこのあたりも機械的に行うようにしなければならない。つまり，最上位1ビットだけを見て商が立つかどうかを調べるところから始めることになる。

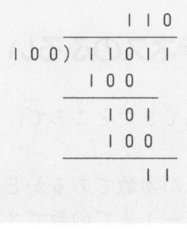
では，実例を見てもらおう。リスト8は32ビット÷32ビットを計算し，商と余りを同時に32ビットで返す除算ルーチンだ。d0÷d1の商がd0，余りがd1に返る。除算の場合はさすがに符号を考慮する必要があるので，符号付き数用のdivs32と無符号用のdivu32の2つのサブルーチンが用意されている。もっとも，divs32のほうは絶対値をとってからdivu32を呼び出し，あとから商と余りの符号を調節するだけの処理しかしていない。肝心なのはdivu32の44〜50行のループだ。

このループではまず被除数d0の上位から1ビット取り出してd2の下位ビットに取り込む（44〜45行）。d2は筆算でいう“被除数のいま注目している上位数桁”を保持する目的で使われる。被除数の最上位ビットを取り出すためにadd命令を使っている点についてはいいだろう。問題はadd命令によってはみ出したビットをどうやってd2に取り込むかだ。d2には直前のビットを処理したときまでの除算の余りが入っているから，これを1桁左シフトしたうえで，最下位桁にいま被除数から取り出した1ビットを繰り込む必要がある。

このようなときにはroxl命令を使うのがひとつの手段だ。あまり使わない命令なので忘れているかもしれないが，roxl命令はオペランド＋ccrのXビットをひと続きのビット列と見なして，ぐるっと左に回転（ローテート）する。ローテートはシフトに似た操作だが，ビット列は輪になっているものとして扱われ，押し出されたビットは反対側から入ってくる。したがって，roxl命令でd2を1ビットローテートすれば，d2を1ビット左シフトするのと同時に，直前の加算命令で生じた繰り上がりであるXビットを最下位ビットに取り込むことができるわけだ。

が，リスト8ではroxl命令の動作をaddx命令で代用している。addxは被加数＋加数＋Xビットを求め

図3　2進法の除算



リスト8　DIV32.S

```
 1: *          d0÷d1=d0...d1
 2:
 3:            .xdef   divs32
 4:            .xdef   divu32
 5: *
 6:            .text
 7:            .even
 8: *
 9: divs32:
10:            tst.l   d1
11:            bmi     div0
12:            beq     retn
13:
14:            tst.l   d0
15:            bpl     udiv                  *正/正 = 正...正
16:
17:            neg.l   d0                    *負/正 = 負...負
18:            bsr     udiv
19:            neg.l   d0
20:            neg.l   d1
21:            rts
22:
23: div0:      neg.l   d1
24:            tst.l   d0
25:            bpl     div1
26:
27:            neg.l   d0                    *負/負 = 正...負
28:            bsr     udiv
29:            neg.l   d1
30:            rts
31:
32: div1:      bsr     udiv                  *正/負 = 負...正
33:            neg.l   d0
34:            rts
35: *
36: divu32:
37:            tst.l   d1                    *除数が0の
38:            beq     retn                  *  除算は未定義
39:
40: udiv:      movem.l d2-d3,-(sp)
41:
42:            moveq.l #0,d2
43:            moveq.l #32-1,d3
44: loop:      add.l   d0,d0                 *被除数の上位1ビットを
45:            addx.l  d2,d2                 *  d2に取り込む
46:            cmp.l   d1,d2                 *除数が引けるなら
47:            bcs     next                  *
48:            addq.b  #1,d0                 *  1を立てて
49:            sub.l   d1,d2                 *  実際に引く
50: next:      dbra    d3,loop               *32ビット分繰り返す
51:
52:            move.l  d2,d1                 *d1.l = 余り
53:                                          *d0.l = 商
54:            movem.l (sp)+,d2-d3
55: retn:      rts
56:
57:            .end
```

リスト9　DIV32.S（最終版：追加部）

```
udiv:      cmpi.l  #$10000,d1            *除数が16ビットに
           bcc     udiv1                 *  収まっている場合は
           divu.w  d1,d0                 *  divuを使う
           bvs     udiv0                 *オーバーフローしたらやり直し

           swap.w  d0                    *
           move.w  d0,d1                 *d1.l = 余り
           clr.w   d0                    *
           swap.w  d0                    *d0.l = 商
           rts

udiv0:     move.l  d2,-(sp)

           moveq.l #0,d2                 *65536進2桁/1桁のつもりで
           swap.w  d0                    *  divuを2回使う
           move.w  d0,d2                 *
           divu.w  d1,d2                 *
           move.w  d2,d0                 *
           swap.w  d0                    *
           move.w  d0,d2                 *
           divu.w  d1,d2                 *

           move.w  d2,d0                 *d0.l = 商
           swap.w  d2
           move.w  d2,d1                 *d1.l = 余り

           move.l  (sp)+,d2
           rts

udiv1:     movem.l d2-d3,-(sp)
```

る命令で，リスト8のように，

　　addx.l　d2,d2

という使い方をすると2倍することによる左シフトとXビットの取り込みが同時に行われることになり，

　　roxl.l　#1,d2

と同じ結果が（少しだけ高速に）得られる。

46〜49行が商を1桁立てる部分だ。いま注目しているd2と除数d1を比べてみて(46行)，d2のほうが大きいか等しいようなら，1を立てて(48行)，d2からd1を引く(49行)，というアルゴリズムどおりの処理をしている。ここにはちょっとした小技がある。被除数を格納したd0の下位ビットを商の格納用に使っているのだ。44行で1ビット左シフトすることで下位ビットが空くことを確認しておこう。このもう被除数としては使われなくなったビットに商を収める。ループが回るにつれて，被除数はどんどん左側に追い出され，下位から商が入り込んでくる。32ビット分処理し終わったときには被除数は完全に追い出されて，商に取って代わられることになる。

ループを抜けた時点でd2には除数未満の被除数の残り，つまり，除算の余りが残る。これを仕様どおりd1に返すために転送して処理完了だ。

せっかくだから，乗算の最終版同様に，実用上の速度を稼ぐ小細工を加えておこう。リスト8の40行をリスト9でごっそり差し替えてもらいたい。この修正により，除数が16ビットで収まる場合には，ループで筆算をまねる代わりにdivuで直接計算するようになる。ここで，除数が16ビットに収まっている場合でも，divuには例のオーバーフローの問題が残っている。リスト9ではこの点を考慮し，試しにdivuで割ってみてオーバーフローした場合は，さらに12行以降の32ビット÷16ビットの専用ルーチンに処理を振り分けるようにした。14〜25行では32ビット÷16ビットを65536進2桁÷1桁とみなし，divu命令2個で目的を達している。

ユークリッドの互除法

除算の応用として，最大公約数と最小公倍数を求めることを考えてみよう。最大公約数の算出にはユークリッドの互除法と呼ばれる古典的なアルゴリズムがある。また，xとyの最大公約数を，

　　GCD（x，y）

最小公倍数を，

　　LCM（x，y）

で表すことにすると，

　　LCM（x，y）＝x×y÷GCD（x，y）

だから，最小公倍数もまたユークリッドの互除法の応用で求めることができる。

ユークリッドの互除法は，

　　x÷y＝q　余り　r

ならば，

　　GCD（x，y）＝GCD（y，r）

という事実に基づいている。

　　x←y

　　y←r

とおいて同様の操作を繰り返すと，いつかはxをyで割った余りrが0になり，そのときのyが求める最大公約数になる[3)]。

これだけ知っていれば最大公約数と最小公倍数を求めるプログラムが書ける。それぞれ，リスト10，リスト11に実例を示しておこう。どちらのサブルーチンも引数をd0，d1で渡すと結果がd0に返る。最大公約数を求めるサブルーチンgcd32では，ループをwhile型にする必要がある点に注意しよう。そうしないと，y＝0のときに0での除算が行われてしまう。y＝0ならx自身が最大公約数だから[4)]，このときはループの中身を一度も実行せずに通り抜けなければならない。また，最小公倍数を求めるlcm32では，乗除算の順序に気をつけたい。xとyを先に掛けてしまうと，LCM（x，y）が32ビットで収まっても乗算の時点でオーバーフローする危険がある。

エラトステネスのふるい

最大公約数が出てきたところで，素数についても触れておこう。

一般に自然数xが素数であるかどうかを調べるためには，2からx－1までの数でxを順に割ってみ

3)　0はどんな整数でも割り切れるから，任意の自然数Aと0との最大公約数はAそのものだ。

4)　正確にいうと，xもyもともに0の場合は最大公約数は存在しないが，gcd32では気にせずに0を返す。

リスト10　GCD32.S

```
*        最大公約数

         .xdef    gcd32
         .xref    divu32
*
         .text
         .even
*
gcd32:
         movem.l d1-d2,-(sp)
         bra      lpent

loop:    move.l   d1,d2                  *d0 = x
                                         *d1 = d2 = y
         jsr      divu32                 *x = y*q+r
                                         *d0 = q
                                         *d1 = r
         move.l   d2,d0                  *x' = y, y' = rとおいて
lpent:   tst.l    d1                     *y = 0になるまで繰り返す
         bne      loop                   *

         movem.l (sp)+,d1-d2
         rts

         .end
```

リスト11　LCM32.S

```
*        最小公倍数

         .xdef    lcm32
         .xref    gcd32
         .xref    mulu32
         .xref    divu32
*
         .text
         .even
*
lcm32:
         movem.l d1-d2,-(sp)

         move.l   d0,d2
         jsr      gcd32
         exg.l    d0,d1
         jsr      divu32
         move.l   d2,d1
         jsr      mulu32

         movem.l (sp)+,d1-d2
         rts

         .end
```

て，どの数でも割り切れないことを確認する。効率を気にするなら，割る数の上限をx－1ではなく$\sqrt{x}$までに限ってもよい。なぜなら，仮に$\sqrt{x}$よりも大きな数Aで割り切れるなら，

$$x = A \times B$$

を満たす数Bが存在することになり，そのような数Bは$\sqrt{x}$よりも小さいからだ。Aで割り切れるなら，Aよりも$\sqrt{x}$よりも小さなBで割ってみた時点で割り切れることがわかるから，処理は$\sqrt{x}$まで進んでこないはずだ。

さらに効率を上げたければ，割る数を素数に限ってもかまわない。ただ，$\sqrt{x}$以下の素数があらかじめわかっていないとどうしようもないので，素数を小さい順に次々に求める場合ぐらいにしかこの方法は使えないだろう。しかも，素数を小さい順に求めるのが目的なら，より効率のよい方法がある。“エラトステネスのふるい”と呼ばれる古典的なアルゴリズムだ。本来の目的よりも，ベンチマークテストで有名かもしれない。

手順は単純だ。

1)　あらかじめ2～nまでの数を列挙しておく。
2)　小さいほうからひとつ取り出してくる（仮にpと置く)。pは素数。
3)　pの倍数である2pや3pはpで割り切れるわけだから素数ではないので消す。
4)　nに達するまで2)，3)を繰り返すと，最後には素数だけが残る。

約数を求めるというのに除算を使わずにすむのがこのアルゴリズムの賢い点だ。pの倍数を求めるところに出てくる乗算も，pを順に加算する操作に置き換えれば消える。

実際のプログラムでは，“2～nまでの数を列挙する”とか，“倍数を消す”とかいう漠然とした処理を具体化する必要があるが，ここは2～nまでの数に対応するフラグの配列を用意することで実現できる。このフラグは初期状態では仮に“素数の印”にしておき，素数の倍数だということがわかった時点で“素数ではないという印”にセットする。なお，2以外の偶数は素数ではない（2で割れる）から，2だけを特別扱いして，3以上の奇数だけを対象に処理を進めたほうがフラグの配列の大きさを節約できるし，速度効率もよい。

実例をリスト12に示す。リスト12は今回作成したプログラムの中では唯一，単独でアセンブル/リンクすると実行形式になるプログラムだ。実行すると11行の記号定数Nで指定した値以下の素数を16個/行で標準出力に書き出す。正確にいうと記号定数Nはフラグの配列の大きさを表し，上述のように奇数だけを処理対象にしたことから，実際にはNの2倍程度の数までの素数が出力される。

ところで，リスト12ではフラグとして各1バイトをあてている。“素数”か“素数ではないか”を表すのには1ビットあればすむから，メモリを8倍も無駄遣いしていることになる。余力のある人は，フラグをビット単位に割り当てるよう改良してみるのもいいだろう。

平方根

最後に整数レベルで平方根を求める方法を見てみよう。もちろん，平方根は整数とは限らないわけだが，ここでは平方根を超えない最大の整数を計算す

リスト12　PRIMES.S

```
 1: *            素数表を出力する
 2:
 3:             .include        doscall.mac
 4:             .include        const.h
 5: *
 6: FPACK       macro   callno
 7:             .dc.w   callno
 8:             .endm
 9: *
10: __IUSING            equ     $fe18   *d0→10進d1桁文字列
11: N                   equ     1499    *最大値(= N*2+1)
12: *
13:             .text
14:             .even
15: *
16: ent:
17:             lea.l   inisp,sp
18:
19:             pea.l   _2str(pc)       *2を出力する
20:             DOS     _PRINT          *
21:
22:             moveq.l #2,d7           *d7 = 改行用のカウンタ
23:
24:             moveq.l #3,d2           *d2 = i = 初期値
25:             lea.l   flags(pc),a1    *a1 = フラグの配列先頭
26:             lea.l   flags+N,a2      *a2 = フラグの配列末尾
27:
28:             moveq.l #5,d1           *出力桁数(IUSING用)
29:             pea.l   numbuf(pc)      *数値→文字列変換バッファ
30:
31: loop1:      tst.b   (a1)            *すでにフラグがセットされていたら
32:             bne     next1           *  iはi未満の数の倍数(∴素数ではない)
33:
34:                          *iは素数
35:             lea.l   numbuf(pc),a0   *iを5桁幅で10進出力
36:             move.l  d2,d0           *
37:             FPACK   __IUSING        *
38:             DOS     _PRINT          *
39:
40:             lea.l   spcmes(pc),a0   *16個出力するごとに改行する
41:             rol.w   #1,d7           *
42:             bcc     prtspc          *
43:             lea.l   crlfms(pc),a0   *
44: prtspc:     pea.l   (a0)            *
45:             DOS     _PRINT          *
46:             addq.l  #4,sp           *
47:
48:             movea.l a1,a3           *iの倍数を消す
49:             bra     next2           *
50: loop2:      st.b    (a3)            *
51: next2:      adda.l  d2,a3           *
52:             cmpa.l  a2,a3           *
53:             bcs     loop2           *
54:
55: next1:      addq.l  #2,d2           *i += 2
56:             addq.l  #1,a1           *(a1) = iに対応するフラグ
57:
58:             cmpa.l  a2,a1           *フラグの配列末尾に達するまで
59:             bcs     loop1           *  繰り返す
60:
61:             pea.l   crlfms(pc)      *改行
62:             DOS     _PRINT          *
63:
64:             DOS     _EXIT
65: *
66: spcmes:     .dc.b   SPACE,0
67: crlfms:     .dc.b   CR,LF,0
68: _2str:      .dc.b   '    2 ',0
69: *                    123456
70: *
71:             .bss
72:             .even
73: *
74: numbuf:     .ds.b   10              *数値→10進文字列変換用
75: flags:      .ds.b   N               *素数かどうかのフラグの配列
76:
77:             .stack
78:             .even
79: *
80:             .ds.l   256
81: inisp:
82:
83:             .end
```

る。2通りの方法を示そう。

第1の比較的単純な方法では，1から始めて3，5，……，$2n-1$までのn個の奇数を足すとn^2になるという事実を利用する。証明はしないが,

$$
\begin{aligned}
&1 &&=1=1^2\\
&1+3 &&=4=2^2\\
&1+3+5 &&=9=3^2\\
&1+3+5+7 &&=16=4^2\\
&1+3+5+7+9 &&=25=5^2\\
& &&\ \vdots
\end{aligned}
$$

をもう何段階か試してみると，納得してもらえるだろう。

ここまできたら，あとは力技だ。開平しようする数から，1，3，5，……，を順に引いていって，負になる直前までに何回引けたかを数えると，その回数が求める平方根になる。プログラム例をリスト13に示す。リスト13のサブルーチンsqrt32はd0を無符号数と見なして，その平方根（を超えない最大の整数）をd0に返す。

リスト13 SQRT32.S

```
*          整数の平方根

           .xdef   sqrt32
*
           .text
           .even
*
sqrt32:
           movem.l d1-d2,-(sp)

           move.l  d0,d2                  *d2 = a
           moveq.l #0,d0                  *d0 = i
           moveq.l #1,d1                  *d1 = (2i-1)
           bra     next

loop:      addq.l  #1,d0                  *i++
           addq.l  #2,d1                  *1,3,5,...,(2x-1)を発生
next:      sub.l   d1,d2                  *引けるだけ引く
           bcc     loop                   *

ret2:      movem.l (sp)+,d1-d2
           rts

           .end
```

リスト14 SQRT32.S(別版)

```
*          整数の平方根

           .xdef   sqrt32
           .xref   divu32
*
           .text
           .even
*
sqrt32:
           movem.l d1-d4,-(sp)

           move.l  d0,d1                  *d1 = 初期近似値(x_0)
           lsr.l   #1,d1                  *    = n/2
           beq     retn                   *n/2 = 0 ならば
                                          *  nは0または1(∴ √n = n)

           move.l  d0,d2                  *d2 = a

           move.l  d1,d3                  *
loop:      move.l  d3,d4                  *d4 = 前々回の近似値(x_{i-1})
           move.l  d1,d3                  *d3 = 前回の近似値(x_i)

           move.l  d2,d0                  *d1 = x_{i+1}
           jsr     divu32                 *    = (x_i+n/x_i)/2
           move.l  d3,d1                  *
           add.l   d0,d1                  *
           lsr.l   #1,d1                  *

           cmp.l   d3,d1                  *収束したら(x_{i+1} = x_i)
           beq     done                   *  それが答え

           cmp.l   d4,d1                  *振動し始めたら(x_{i+1} = x_{i-1})
           bne     loop                   *
           cmp.l   d3,d4                  *  x_{i+1}とx_iの小さい方が答え
           bcs     done                   *
           move.l  d3,d1                  *

done:      move.l  d1,d0                  *d0 = √n

retn:      movem.l (sp)+,d1-d4
           rts

           .end
```

第2の方法は適当な近似値にじわじわと修正量を加えて真の値に近づけていくという方法だ。まず，nの平方根の近似値x_0を用意し，x_0と$\sqrt{n}$の正しい値の差をcと置く。

$$c=\sqrt{n}-x_0$$

すると，

$$
\begin{aligned}
n&=(x_0+c)^2\\
&=x_0^2+2cx_0+c^2
\end{aligned}
$$

だ。このcがそのまま解ければよいのだが，そのためには平方根が必要でヤブヘビになる。しかしc^2はほかの項よりも十分小さいと予想されるので無視すると，

$$n\fallingdotseq x_0^2+2cx_0$$

となり，この式は簡単に解けて，

$$c=\frac{1}{2}\left(\frac{n}{x_0}-x_0\right)$$

が得られる。途中で微小項を略してしまったために得られたcは$\sqrt{n}$とx_0の差そのものではないが，x_0+cはx_0よりも$\sqrt{n}$に近い，よりよい近似値だ。そこで，

$$x_1+=x_0+c$$

と置いて同様の操作を繰り返すと，x_iはしだいに$\sqrt{n}$の真の値に近づいていき，cの絶対値が十分小さくなったとき，x_iに$\sqrt{n}$が求まっていると考えられる。

このアルゴリズムは除算を使うので，第1の方法よりも遅くなりそうに見えるかもしれない。実際，nが小さいときには第1の方法のほうがずいぶん速い。しかし，nがある程度より大きくなると立場は逆転する。こだわりたければ，nの大小に応じて2つの方法を使い分けるとよいだろう。

リスト14は第2のアルゴリズムを使って平方根を求めるsqrt32の別版だ。初期近似値は非負の数ならいくつであってもよいのだが，ここでは開平しようとする数nの2分の1にしてある。

このプログラムではループの脱出条件が重要なポイントだ。一見，整数レベルで計算するのだからcの絶対値はやがて0になるように思える。ところが実際には除算の段階での切り捨ての影響もあって，1，−1，1，……，と振動することがある。そこで，リスト14ではループの終了判定にcは使わず，近似値x_iを2段階前の分まで保存しておいて，きっちり収束するか，振動を始めたことを確認した時点でループを抜けるようにした。なお，振動し始めた場合，$\sqrt{n}$の真の値は2つの近似値の間にある。いま求めたいのは$\sqrt{n}$を超えない最大の整数だから，小さいほうを答えにすればよい。

というあたりで次回へと続く。次回は32ビットなんていうレベルではない極端に大きな数を取り扱う。100どころか1000の階乗が正確にいくつかを求められるような多倍長整数演算ルーチンを作成することになりそうだ。

THE USER'S WORKS

●BLUE WINGS

以前，このコーナーで紹介したこともある(ULTIMATE MAGIC）サークルO/S Software制作の縦スクロールシューティングゲーム。グラフィック，音楽，ゲーム性と3拍子揃った秀作だ。

●

「第47特殊戦術航空隊，通称BLUE WINGS。エキスパートパイロットばかりで構成された超エリート部隊である」基本設定はなんとなくメタルサイトに似ている。

6面＋α，各面ボスキャラつきという，縦スクロールシューティングとしてはオーソドックスな構成である。

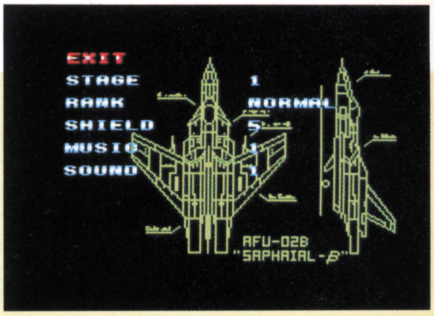
基地からの発進シーンからゲームは始まる。1面の背景は海上から密林，砂漠と変わっていく。2面以降は，なんとなくイメージファイト面（シムシティー面ともいう），エアデュエル面，空中要塞面，宇宙面を経て敵母星面，そして最終面へと至る。各面ともグラフィックの描き込みがよく行われている。画面を見ればわかるとおり，グラフィックはアイレム系の影響が色濃い。まあ，これはよいことなのであろう。

パワーアップシステムは特殊兵器の選択によるもので，味方機の落としていくカプセル（一定時間で種類が変わる）をタイミングよく取っていく。

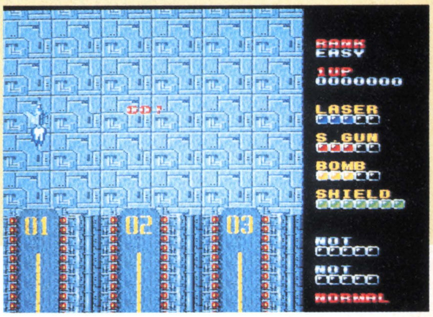
拾ったアイテム（4種類）のうち，レーザー，ショットガン，ボムを選択する。最初から3個ずつ（最大5個）持っており，いつでも使える状態にあるため，ゲーム再開時でも極端なパワーダウンはない。

ただし，それぞれ，一定量使用するとなくなってしまうので，ボス以外は通常弾でしのいで後半に備えたほうがよい。通常弾も2段階パワーアップ可能だ。面白いのは

タイトル画面だ



コンフィグレーションモードもある



さあ，発進だ！

通常弾のパワーアップがシールド回復（1段）を兼ねていることだろう。カプセルをパワーアップアイテムとして使うか，シールド回復に使うかが選択できる。

さらに2種類のカプセルを使用した複合パワーアップが可能。このゲームの最大の特徴といえるだろう。これにより，全部で9とおりのパワーアップ体系となる。レーザーを重ねたスーパーレーザーやスーパーボム（B＋B）などの強力兵器は対ボスキャラ用，ショットガン（3方向弾）やスプレッド（S＋S：3方向誘導弾）などは後半面の激戦区で効果を発揮する。

操作性は普通程度。自機の動きが多少カクカクしているが，ほとんど問題はない。ちなみに連射装置は使うだけ無駄（自動連射)。音楽もノリがよい。

敵の中ボスやボスキャラがかなり固いのだが，撃ってもどの程度のダメージを与えたのかがわからないので，いまひとつ手応えに欠ける。その点だけが残念だった。

気になるゲームバランスだが，HARDは人間のやるもんじゃないとして，EASY，NORMALではバランスもよいと思う。

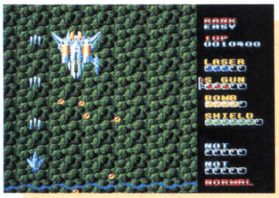
面ごとにだんだんと敵の攻撃が激化していく。特に要塞内でのサーチレーザー砲台と敵機の複合攻撃がきつい。それでも基本的に敵の弾は避けられる。バラバラと画面中が弾だらけになっても落ち着いていればなんとか抜けられる密度である。特にボスキャラでは弾避けが熱い。

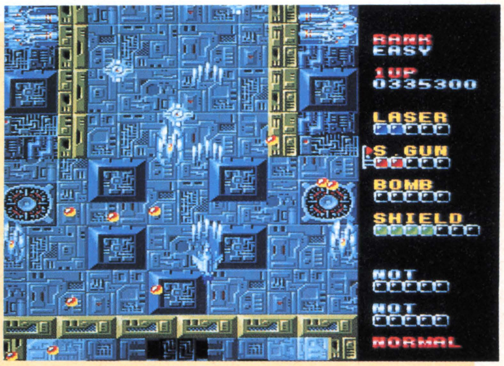
入手方法

このゲームを入手希望の方は，無記名の郵便小為替，返信先住所を明記したタックシール（または紙に書いたもの）を用意し，封書にて連絡してほしい。なお，価格は送料込みで1,500円。あて先は下記まで。

〒346　埼玉県久喜市南2-2-7

大森方 O/S software





各面のようす。背景グラフィックの描き込みはなかなかのもの。敵弾の視認性もよい

［第16回］
仮想生物

寺尾　響子

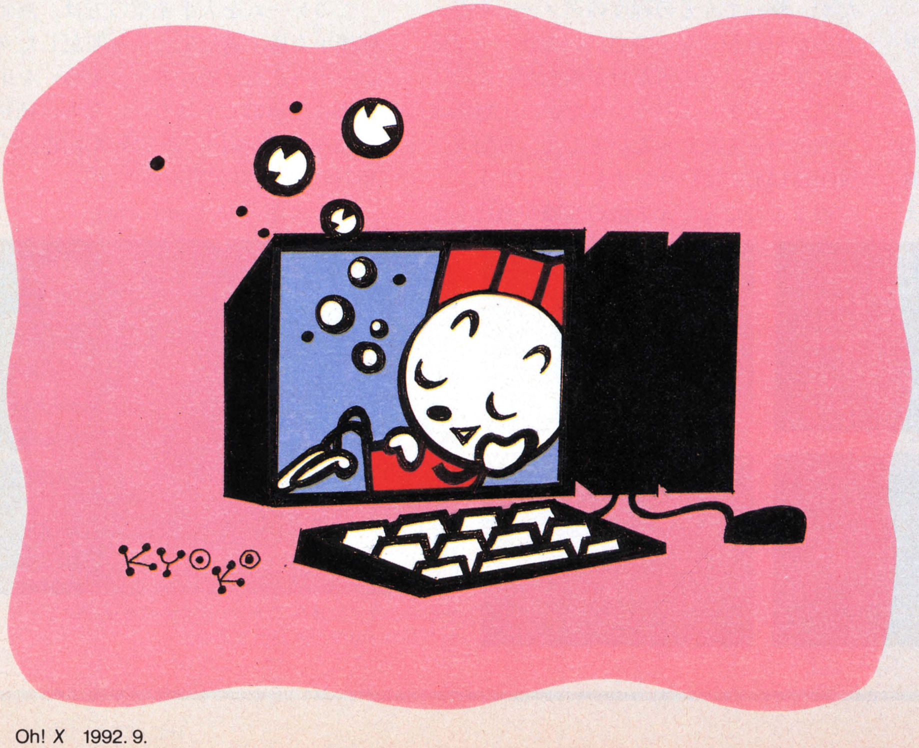
響子inCGわ〜るど

暗闇のなかで，ぽつんとなにかが発光し，生まれていきます。
ねずみのようであって，ねずみではない……
ロボットのようであって，ロボットではない……
鳥のようであって，鳥ではない……
人間のようであって，人間ではない……
この世のものではない，異形のものたち。もうひとつの世界，ファンタジーワールドの住人たち。

神と愛情

魂のある，自分以外の生き物を造ってみたいという欲求は誰しももっているようです。
「仮想生物[1]って，どう思う？」とたずねると，たいていのひとはとても興味を示して，こんなふうに答えてくれました。
「この世には存在しない，なにか別のものをこしらえて，命を吹き込んで動かしてみたい」と。

ずいぶんいろいろなひとに聞いてみましたが，男のひとと女のひとでは，仮想生物を造りたいという動機が微妙に違うようです。
「自分の人生の先が見えているような，いまの環境とは全然違う環境を造って，自分に似たものがどんなふうになるか見てみたいよ」
「僕は神になりたい」
「創造主になって，世界を思いどおりに支配してみたいぜ，俺は」
なるほどなるほど，ポピュラスやシムシティーやシムアースがよく売れるわけだ。うん。
女のひとは，
「永遠に裏切られることのない，子供を育ててみたい」
「死ななくて，成長もしない，かわいいペットがほしいなあ」
「私だけを愛してくれる動物（人間も含まれてい



ますね，これは絶対に）がいたらいいのに」

「ずっと愛情を注ぐことのできる，なにかを造りたい」[2]

ふむふむ，納得。自分自身が女性なので，よくわかります。

さて，私自身はどうなのかしら。

「誰の心のなかにもある，そして自分の心のなかにもある，もうひとつの世界。現実の裏側に潜んでいるパラレルワールドをこの目で見てみたい」というところでしょうか。すこし現実逃避なのかもしれません。

*　　*　　*

実は，いまゲームを作っています。

この世には存在しない３次元空間をコンピュータのなかにこしらえて，そこをさまざまなキャラクターたちが自分の意思をもって勝手に動き回っている。

プレイヤーはひとつのキャラクターになって，その世界に参加する，というようなゲームです。昔からある，飛び出す絵本のようにできればと考えています。

心が休まる空間を，ロクハチのなかにそっと置いておける。そんなゲームにしてみたいと思っています。

---

1)　仮想生物と聞くと，映画「ブレードランナー」（原作はP.K.ディックの「人造人間は電気羊の夢を見るか？」）のレプリカントを思い出します。

2)　女性の仮想生物に対するイメージは，小説「ネットワークベイビー」（一色伸幸・原作，田村章・著，太田出版）を読むとよくわかります。これは，NHKスペシャル「ニューウェーブドラマシリーズ」（1990年５月１日放映）を小説化したものです。

MIRAGE System Model Stuff(2)

Tan Akihiko 丹　明彦

モジュール拡張型3D CGシステム「MIRAGE System」の第1弾として，Model Stuffが発売されました。モデラ重視とはいってもレイトレーサとしての基本機能は十分。使い勝手はどうでしょうか?

7月号で概要をお伝えしたMIRAGEであるが，製品の発売時期と原稿の締め切りとが悪いぐあいにずれたために，1カ月以上遅れての追加レビューとなってしまった。もうお使いになっている人も多いことだろう。

というわけで，今回は製品バージョンの評価ということになる。前回はレンダリング部分抜きでのレビューだったが，今回はちゃんとサンプルも作ってみた。

＊　＊　＊

いきなり感想を述べてしまおう。X68000にもやっと本格的に使う気になるレイトレ用モデラが出てきたといったところである。僕がOh!Xでレイトレのレビューを始めて以来，もっとも複雑なサンプルを作ったという事実がそれを物語っている。

レンダラの機能としては特に目新しいところはない。サイクロンとほとんど同レベルといえる。

全体として，既存のレイトレソフトの機能は踏襲したうえで，そこそこ使い勝手のよい環境を用意したというレベル。海外機種も考慮した他機種の状況を眺めると，この程度はもはや当たり前ともいえる。逆にいえば，いままではその程度も用意されていなかったのだ。

X68000用5"2HD2枚組　29,800円(税別)
メディックス　☎03(3950)2222

すべてはここからだ。まだまだ改良の余地はあるし，バグも多いので，早々にバージョンアップを期待したい。

システム構成

MIRAGEシステムは，一度起動するとモデリングからレンダリングまでをすべてその上で行える環境である(少なくとも見掛け上は)。独自のウィンドウ(?)システムだし，マルチタスクでもないが，お客様気分で気軽に扱える。

MIRAGEシステムを起動すると，「MIRAGEシェル」と呼ばれるウィンドウが開く。その中には10数個のアイコンが並んでおり，それをクリックすることで以下のようなサブプログラム群を呼び出すようになっている。

●ロード
　モデリングデータを読み込む。

●セーブ
　モデリングデータを保存する。

●モデラ
　形状デザインを行う。

●アトリビュータ
　属性デザインを行う。

●レンダラ
　作成したデータのレンダリングを行う。画面の縦横比を補正するアスペクト比は1.0と0.8から選べる。

●画像ビュア
　24ビットフルカラー画像を画面に表示する。65536色に落とすとき，単純に量子化するものとディザ処理を加えるものとがある。

●マップカッタ
　テクスチャマッピングやバンプマッピングなどに使うマップデータを切り出す。まず元画像をロードし，マウスを使って望みの大きさに切り取る。

●Z'sSTAFF
　ペイントツールZ'sSTAFFをMIRAGEから利用できる。マッピングデータを作るのに使える。無論，Z'sSTAFFがないと動作しない。

●Z'sTRIPHONY
　ポリゴンモデリングツールZ'sTRIPHONY DIGITAL CRAFTをMIRAGEから利用できる。MIRAGEのレンダラはポリゴンをサポートしている。モデラは簡易ポリゴンのみサポートしている。無論，Z'sTRIPHONYが必要である。

●POL_TRI
　Z'sTRIPHONYでデザインしたデータをMIRAGEの形式にコンバートする。あらかじめTRIPHONY側では「テキスト形式」でセーブしておくこと。

＊　＊　＊

作法はメニュー形式の域を出ていないが，選択肢が目の前にぶら下がっていてマウスでつつけばいいというのは，楽である。

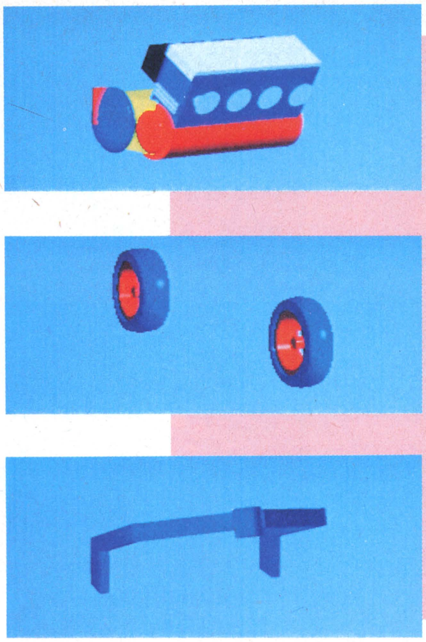
公開仕様ではないが，このメニューはカスタマイズできる。定義ファイル(shell.cmd)が通常のテキスト形式で書いてあり，親切なコメントも入っているので，簡単に書き換えて使える。僕はZ'sSTAFFの代わりにZ's-EXを呼び出すようにしているし，MATIERも組み込んでみた(MATIERが発売されたら，MIRAGEでも正式サポートしてほしい。これほどレイトレ向きのツールはない)。それと，COMMAND.Xがチャイルドプロセスで呼べないのは不便なので，そのエントリも追加した。

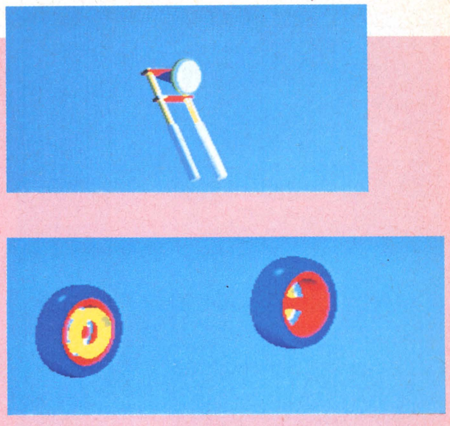
モデリング部の構成

CGツールのなかでいちばんユーザーとの接点が多く重要な部分はモデリング部分である。上でも述べたとおり，MIRAGEのモデリング部は，形状定義と属性定義を行う部分がなぜか別になっている。モデラとアトリビュータを独立にバージョンアップするためだろうが，MIRAGEシステムは形

状と属性の結合が希薄で，ちょっと困ったこともときどき起こる。
ユーザーはモデリング作業のあいだ，モデラとアトリビュータを頻繁に行き来する。その場合いちいちMIRAGEシェルに戻るのは面倒だという配慮からか，モデラにはアトリビュータに，アトリビュータにはモデラに，それぞれダイレクトに移行するアイコンが用意されている。タスクスイッチの感覚でモデリング作業を進めることができる。

モデリングデータをセーブ/ロードするときはいったんシェルに戻る必要がある。モデラとアトリビュータが分かれているため，共通のモデリングデータをシェルが管理する形になっている。





まず，オートバイの各構成単位に分けてモデリングする。これは，それぞれを試しにレンダリングしたもの。モデラ画面でも入り組んだ論理演算部分などは形状の確認ができないので，このようにレンダリングして形状確認を行う。エディタに論理演算表示がほしかったところだ

マウスオペレーションのモデラ

まずはモデラから見てみよう。

7月号でお伝えしたとおり，MIRAGE Model Stuffはモデラを第一の売りにしている。プリミティブのアイコンをクリックすると作業画面にプリミティブが出現する。それをマウスでつかんで移動したり回転したりリサイズ(大きさを変える)したりする。初めに数値を入力することを強要する旧来のモデラとは一線を画す。とりあえず画面を見ながら作業が進められるのはうれしい。今回のサンプルも，設計図なしで作っている。さすがに寸法は狂ったが。

Model Stuffで使えるプリミティブは以下のとおり。

- ●直方体
- ●楕円体
- ●円柱
- ●円錐
- ●一葉双曲面
- ●二葉双曲面(の片方)
- ●ポリゴン
- ●点光源
- ●平行光源
- ●スポット光源

レイトレのモデラとしては無難な線である。例によって，これらを積木細工の要領で組み立て，望みの形を作る。

ポリゴンは平面多角形を掃引する簡易ポリゴンと，任意の形状を定義できるポリゴンが扱える。ただし，現バージョンのMIRAGEの中だけで作れるのは簡易ポリゴンのみ。ちゃんとしたポリゴンは，Z'sTRIPHONYでデザインしておいたものを前述のコンバータにかけて，モデラから読み込まなくてはならない。

この機能の使い勝手ははっきりいってよくない。Z'sTRIPHONY側でいう「オブジェクト」単位でしか読み込めないこと，色もそのオブジェクト単位でしかつけられないこと(Z'sTRIPHONYではポリゴン単位で色をつけられるので，1色1オブジェクトになるように気を使ってモデリングする必要がある)。読み込んだ時点で色の情報が消えてしまうこと，などがその理由。

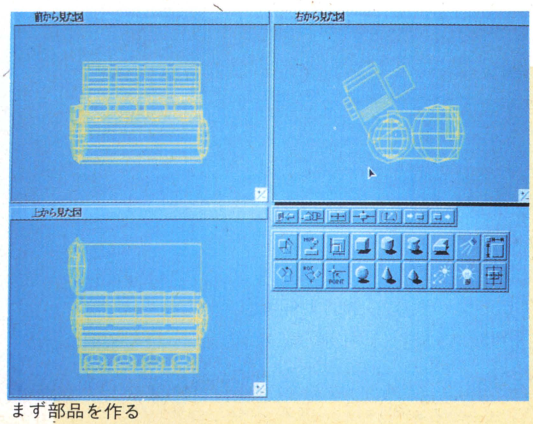
これらはサイクロンに同等な機能がついたときにも指摘した覚えがある。コンバートの操作があまりに煩雑なので(マウスオペレーションで若干救われてはいるが)，レンダリングしてモデリングをやり直すというループ作業を成し遂げる気力がそがれてしまう。色はデフォルトで元の色に近いものをつけ，シーンを一括して読み込むくらいはしてほしかった。

それでも，ポリゴンで自由形状が扱えるのは魅力である。事実，サンプルに「真っ赤なF1マシン」を作った(ではなぜここに写真がないのかというと，レンダラが不調で，まともにレンダリングできなかったからである)。

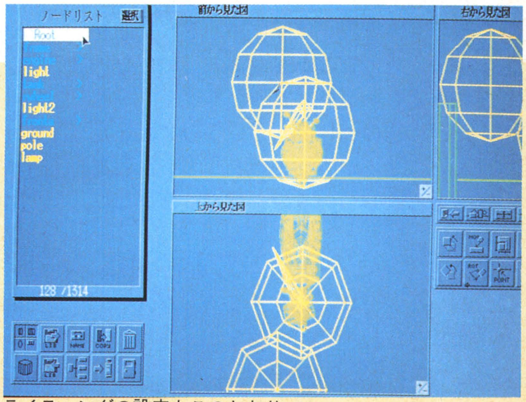
もちろん論理演算もサポートしている。2つのプリミティブの共通部分をとったり，差をとったりして，もともとは単純な形のプリミティブで複雑な形を表現する機能である。

論理演算はグループ化の際に指定する。グループ化したものをさらにグループ化していくこともできるので，複雑なツリー構造が構成できる。ブラウザっぽいノードリストのおかげで，奥深い階層に埋もれているプリミティブにも簡単にアクセスできる。

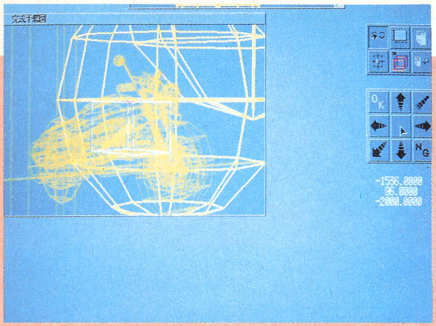
光源をプリミティブと並べて書いたのには意図がある。光源も通常のプリミティブと同様に，移動したり回転したり，さらにはグループ化したりもできる(もちろんマウスオペレーション)。これはけっこう偉大なことかもしれない。車のヘッドライトの

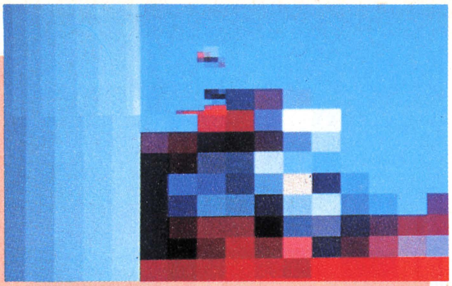


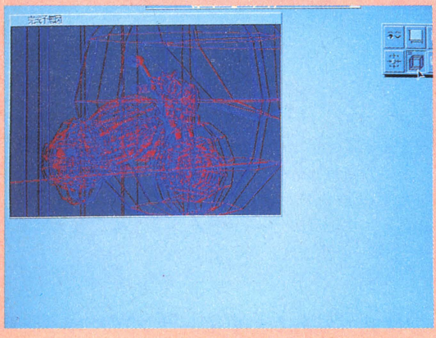
まず部品を作る

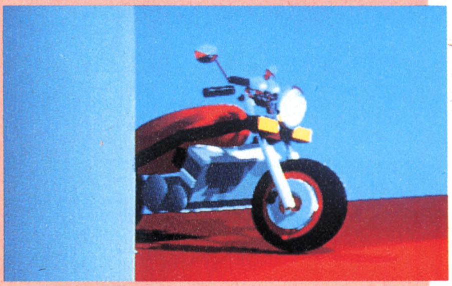


ライティングの設定もこのとおり









モデリングの様子。ワイヤーフレームの基本画面に3Dメガネ用の赤青立体表示画面。レンダリングは再帰的にだんだん細かくなっていく

位置に光源を組み込んでグループ化しておけば，車を動かしても光源が置いていかれない。レイトレでは，これは当たり前なことでもなんでもない。

グループ化した部品をライブラリとして活用できる。グループを選択してセーブし，再利用できる。今回僕がオートバイをデザインするにあたっては，エンジン，フロントフォーク，フレーム，前後輪，ブレーキ，タンクなどを単位として別々にデザインし，ライブラリとしてセーブしておいて，最後に組み上げるというアプローチをとった。さらにフロントフォークと前輪もグループ化し（階層的にグループ化した），ハンドルを切るとフロント全体がつられて動くようにしている。

ライブラリを使っていて気になったのは，ライブラリをよせ集めたときに色化けすることがあること。今回も，エンジンなどの色がいつの間にか変わっていて驚いた。システム構成やデータ構造などを考えるとこうなるのも不思議ではないが，それは開発者の事情。色の管理がきちんとしていないことには変わりない。ライブラリをロードするたびに色のつけ直しというのではいただけない。せっかくライブラリという便利な概念を導入しているのに，もったいないことだ。

編集用のコマンドとしては，移動・回転・拡大縮小などがある。プリミティブAをプリミティブB上にくっつけたり，プリミティブCを点Pの周りに回転させたりといった器用なコマンドも備えている。いずれにせよ，マウスでバウンディングボックスをつかんで見ながら動かせる。形状をテキストエディタで記述するタイプのものだと，回転の方向や角度をうっかり間違えて，“ゼリーの開き”ができてしまったりするのだが，残念ながらMIRAGEではそのような楽しいことは起こらない。

モデリングは基本的に3面図で行うが，構図を決めるときは透視図モードに移行する。ここで視点と注視点と画角を決める。レイトレでは伝統的な作法だが，座標でやらないZ'sTRIPHONY的なやり方のほうが数字を意識しないですむ。

透視図の（三面図もだけど）描画はとても遅く，忍耐が必要。

おまけで3Dメガネもついている。青と赤のセロハンが入ったあれだ。立体視アイコンをクリックすると，赤と青の線で描画する。3Dメガネをかけて見ると，なんとなく立体に見えないこともない。

＊　＊　＊

かなり力の入ったモデラである。だが不満がなくもない。いや，まじめに使おうとすると不満な箇所はかなりある。

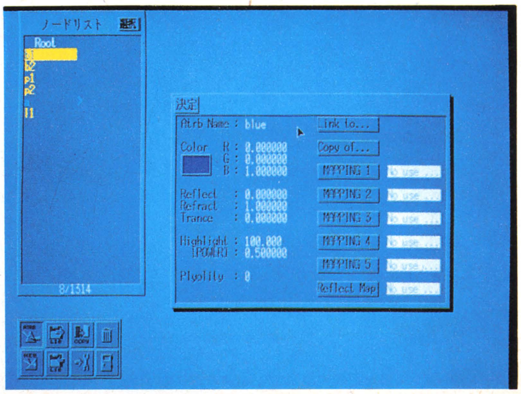
モデラの力を試そうと思ったら，「宇宙戦車」というキャプションがつくようなサンプルを作っていてはダメである。具体的な目標を設定し，それがモデラで表現できるか挑戦してみるのだ。今回はオートバイを作ってみたが，なかなかハードだった。そのハードさのなかには，モデラの仕様をもう少しなんとかすれば解消できる種類のハードさがある。それをここで指摘しておきたい。

まず，遅い。ワイヤーフレーム表示にもかかわらず遅い。16MHzマシンでも十分遅い。なにかするたびにいちいち座標計算をしているという感じの遅さである。モデルが複雑化してくると，たったひとつの操作のたびに30秒も1分も待たされることになる。PC-9801の最新機種ではないのだ。重い計算はできるだけ避けたい。DōGAのCADとまではいわないが，少なくとも気持ちよく作業できるくらいのレスポンスはほしいところ。

また，表示がゴチャゴチャしてくるので作業がやりにくくなる。今回くらいの複雑さのモデルを作っていると，線が重なって潰れるので細かい調整がしにくくなる。画面描きかえにかかる時間もどんどん長くなっていく。オブジェクトを選択的に表示・編集できる機構を用意するとか，論理演算をモデリング中に計算して，その結果だけ表示するとかしてほしい。

論理演算の結果がレンダリングするまでわからないというのも困りものである。モデリング中にわかれば，作業の見通しもかなりよくなるだろう。別に精密な計算はいらない。いいかげんでもいいから，だいたいの形がわかることが大切。今回のサンプルのオートバイでも，特にタイヤ部分は論理演算に使ったプリミティブを全部ナマのままで表示するので，論理演算した結果のタイヤの大きさはまったくわからなくなった。モデリングの透視図と，レンダリングした結果を見比べてほしい。

正確な入力がしづらい。マウスはアバウトな入力を許容する入力機器である。作業スペースの中で，「これくらいの大きさの物体を作る」という用途には向いている。半面，画面の解像度以下の移動はできないので精度に限界がある。手ぶれによる誤差に



これがアトリビュータの画面

も弱い。正確な入力には不向きである。拡大率などの数値はキーボードから指定することもできるが，今回はその回数が妙に多かった。オートバイのように左右対称のものは，左右で一組になっているものを正確に動かす必要があるのだが，マウスではそれは事実上不可能。ドット単位の手ぶれが出るので，使っていてストレスがたまる。せめてオートグリッドがあればよかった。

アンドゥはない。代わりに，一時的に作業状態をしまい込むメモリを用意している。記憶アイコンをクリックして記憶し，あとで呼び出すことができる。ただ，アンドゥは「あ，しまった」というミスに対処することに使われるケースがほとんどで，マメに，しかも手動で記憶アイコンを押すというのはあまり一般的でない。この機能は，あくまで取り返しのつかないような大きな変更の前に使うもののようだ。

＊　＊　＊

いろいろと変な作法もあるし癖も強いが，安定してきたら使う気になるモデラである。

C-TRACEのようにデータをテキストエディタで書くタイプのものは，モデリングデータのすべてをコントロールできるし，個人的には好きだ。が，座標や角度をそれほど気にしなくてすむMIRAGEのモデラは，わりとサクサク書けるので，プリミティブをたくさん使うのがおっくうにならない。これはけっこう重要なことだ。

アトリビュータ

モデラ以外は駆け足で進める。

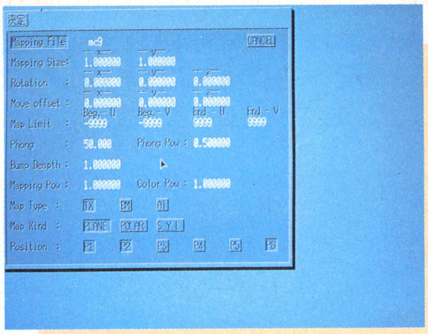
アトリビュータは簡単にいえば色をつけるツールである。単体のプリミティブまたはグループに対して色指定ができる。

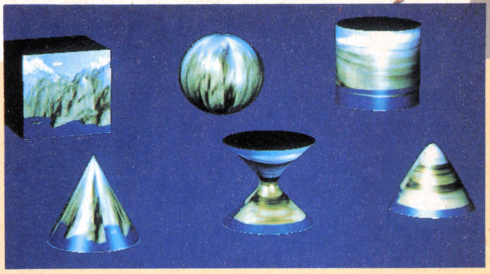
マッピングを設定するのもアトリビュータの仕事だ。テクスチャマッピング/バンプマッピング/アトリビュートマッピングの3種類のマッピングをサポートしている。使える座標系は直交座標と球座標と円筒座標の3種類。

設定したアトリビュートはライブラリ化して再利用できる。

インタフェイスは，かなり不満。今回はモデラのオマケとはいえ，RGB値くらいスライドボリュームで設定させてほしい。こんなものはキーボードで入れさせてはいけない。質感のプレビューもできない。たとえばMATIERではハイライトも含めた質感のシミュレーションをやっているのだ。

MIRAGEシリーズとして発売予定の高性能アトリビュータに期待しよう。





マッピングのいろいろ。マッピング画像は引き延ばされても補間された美しい仕上がりになっている。扱えるマッピングの種類も非常に充実している

レンダラ

レンダリング機能は冒頭に書いたようにオーソドックスなもの。

レンダリングは比較的高速。ボクセル分割を使っているので破滅的に遅くなることはまずない。数値演算プロセッサを直接ドライブする形式のレンダラではないが，そのタイプも計画されているらしい。

アンチエリアシングあり。マッピングの品質もよい。

うれしいのは，画面上に最初は粗く描き，だんだん細かくしていく方式である。これだと早いうちに形が確認できる。上からちまちまと1ドットずつ描いていくやり方よりも精神衛生上いい。

レンダリング中にESCキーを押せば中断することができる。そして，後日レンダラを起動すると，自動的にその続きをやってくれる。これももはや常識的な機能だがうれしい。2日間X68000を連続運転するよりは，寝ているあいだの1日8時間ずつをレンダリングに当てて6日間かけるほうがムダな時間が少なくてすむ。

画像は必ずファイルに落とす(画像ファイルを見て，どこからレンダリングを再開するか継続するかを決めている)。MIRAGEは，画像に限らず，特にセーブやロードというアクションを起こさなくても，データを頻繁にディスクとやりとりする。このため，ハードディスクはほぼ必需品となる。

おわりに

モデラの操作性は創造意欲に大きく影響する。その意味でMIRAGEの発売は喜ばしいことである。

しかし，表現できる形状の自由度はまだまだ低い。今回のサンプルのオートバイは，たまたま構成部品がMIRAGEのプリミティブで表現可能なものばかりだったので，誰でもオートバイと認識できる形に仕上がっている。が，2次曲面とフラットシェーディングのポリゴンだけでは，トルネードは作れないのだ。

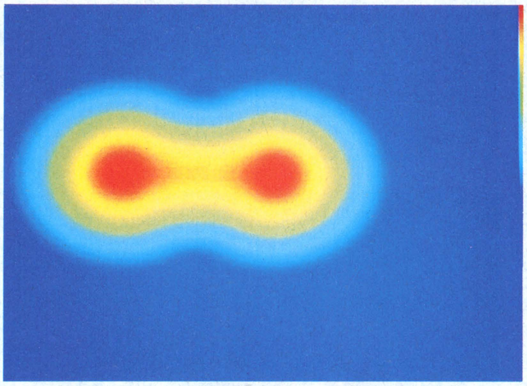
DōGAはポリゴンをベースにしたシステムだが，スムーズシェーディングをサポートしている(それもかなり初期から)。そのため形状の表現力はとても高く，工夫すればレイトレーシングに迫る質感も表現できるとあって，多くのファンを獲得している。もちろん描画の速さも大きな魅力である。

いつかはレイトレーシングで自由曲面を扱ってみたい。ベジェでもスプラインでもいいのだが，乗用車やレーシングカーのデザインをやってみたくなるようなプリミティブがひとつほしいのである。

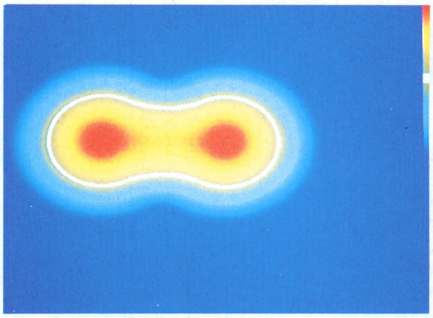
ともあれ，場所はできた。とりあえずは，安心して使えるようにしないとならないだろうが，それを終えたら新機軸に挑戦していただきたいものだ。そしてMIRAGEにはそれをやってくれそうな雰囲気がある。

数値演算高速化の世界

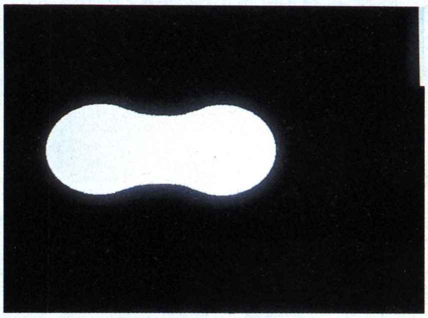
ここには，現在X68000で扱うことのできる数値演算を使って，それぞれ演算高速化へのアプローチをかけた結果が集結している。徹底したモデル化とコーディングによって実現した2次元メタボール，AFPPを使った3Dグラフィックなどで得られた成果は大きいはずだ。そして，いうまでもなくディスプレイには演算結果しか現れない。画面の陰には，CPUやI/Oポートに接続されたAFPP，数値演算プロセッサががんばっていることを忘れないでいてほしい。



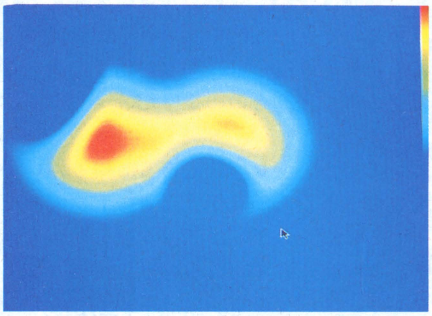
互いに影響するように2個のメタボールを配置



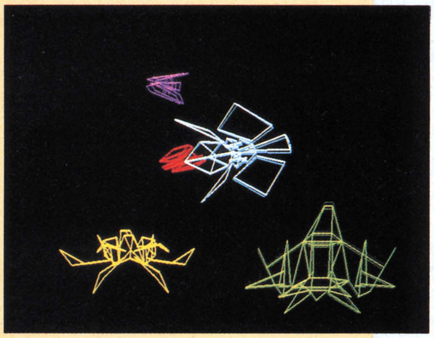
しきい値枠の表示

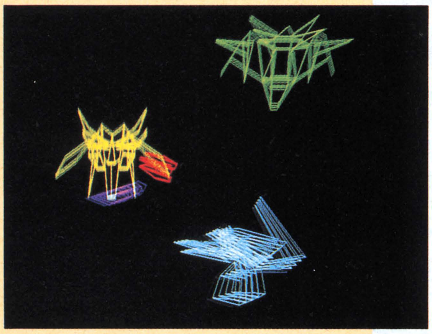


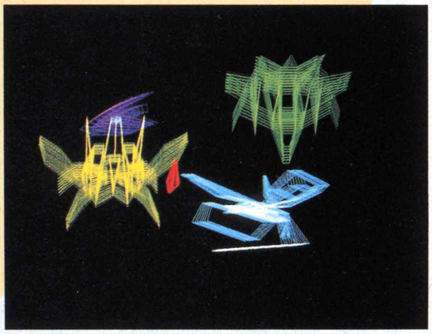
しきい値内部はひょうたん型になっている



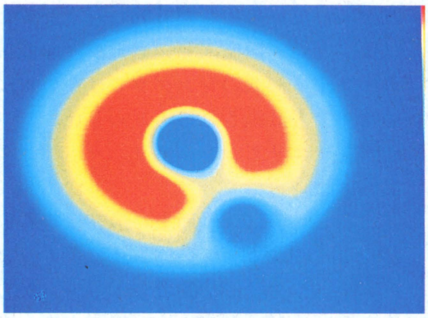
負のメタボール2個を使って変形してみた

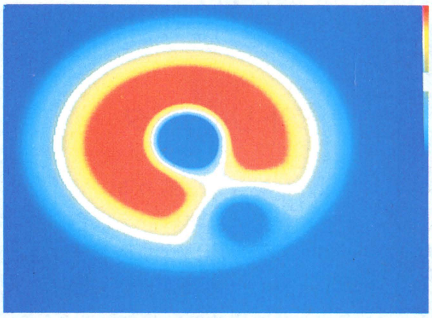




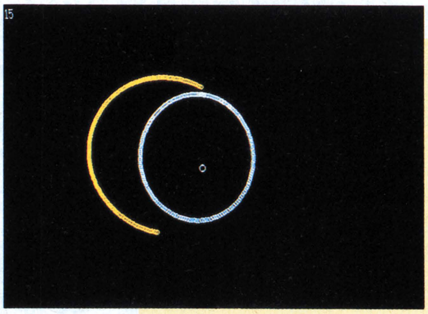


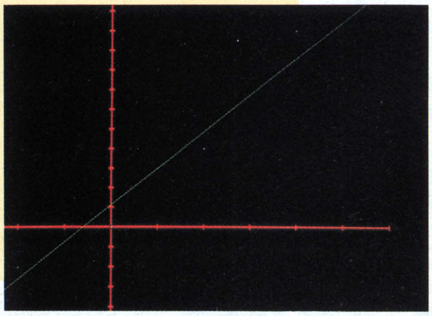
基本のFLOAT2.Xを使ってみたもの（写真上）と実数演算部にAFPPを使用したもの（写真中）さらにラインを高速版に変えてAFPPを使ったもの（写真下）の速度比較





画面中央にメタボール2個を置き，中心を負のメタボールで削ったもの（写真右）。しきい値枠はこんな感じ（写真左）





画面は地味だが2枚の数値演算プロセッサボードががんばっている惑星運航シミュレーション（写真左）。最小2乗法によって求めた1次式のグラフ（写真右）

［特集］

数値演算の熱い逆襲

いうまでもなく現在のコンピュータは，ソフトウェアとハードウェアで構成されている。何かをやるためには，ハードウェア，ソフトウェアの２方向からアプローチをかけることができる。

ソフトウェアを改良することによって高速化を図る。

これは，プログラマの技術しだいでどうにでもなるはず。最適なアルゴリズムを導き出し，最適なコーディングをする。重い，遅い，不可能だ，こんなことばかり嘆いていても進歩はない。与えられたシステム以上のものをソフトウェアで実現する，不可能を可能にする精神だ。

特定のハードウェアを付加することで高速化を図る。

ソフトウェアとはまた違うアプローチであるし，別段非難する理由もない正当な方法である。市販されたボードを差すだけで使い勝手がよくなる，ユーザーからしてみればこれほど魅力的な言葉は存在しない。

しかし，それらのハードウェアもフルに活用するとなると，ソフトウェアの存在を忘れることができない。ハードウェアもソフトウェアからのアプローチがあって，初めて最高のパフォーマンスを見せることができる。たとえ単一の命令が高速化されても，全体の処理がもたついては無意味なのである。

そして，高速化をするためには避けて通れない数値演算。ターゲットが複雑になればなるほど，深く，重くのしかかってくる非常に厄介なものだ。

これからも，どんどん複雑な事象を扱わなくてはならない場合がやってくる。現在でも，たかが数値演算といっていられないところまできているかもしれない。そのようなときに，私たちユーザーはどのように対処すればいいのだろうか。数値演算の熱い逆襲はとどまることを知らないだろう。

CONTENTS

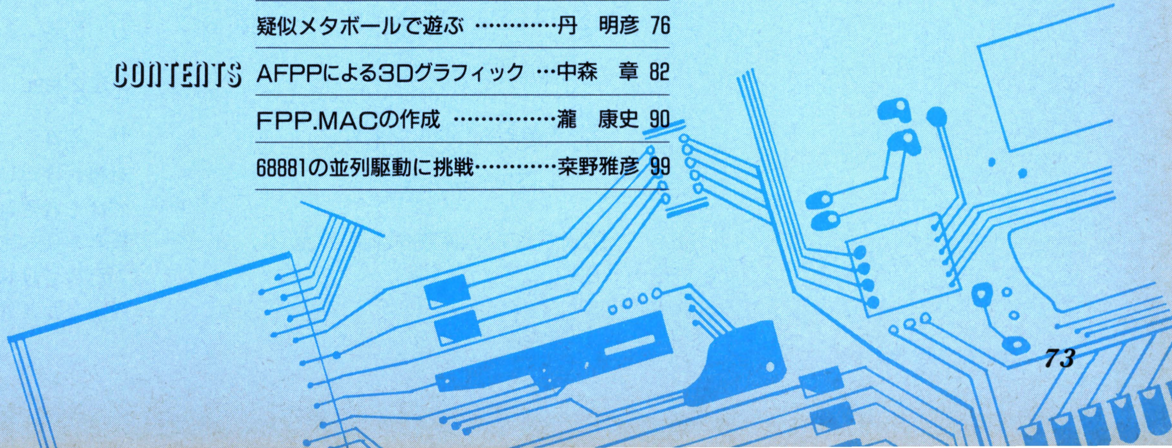
夏休みの最小２乗法……………御木徳高　74

疑似メタボールで遊ぶ…………丹　明彦　76

AFPPによる3Dグラフィック…中森　章　82

FPP.MACの作成……………瀧　康史　90

68881の並列駆動に挑戦…………籴野雅彦　99



微積分をシミュレートする

夏休みの最小2乗法

Miki Tokutaka　御木 徳高

まず，X-BASICを使って計算をさせてみます。テーマは最小2乗法と積分のシミュレート。ちょっとだけ数学の知識を要求しますが，手軽に入力できますのでデータを変えながら遊んでみてください。

最小2乗法

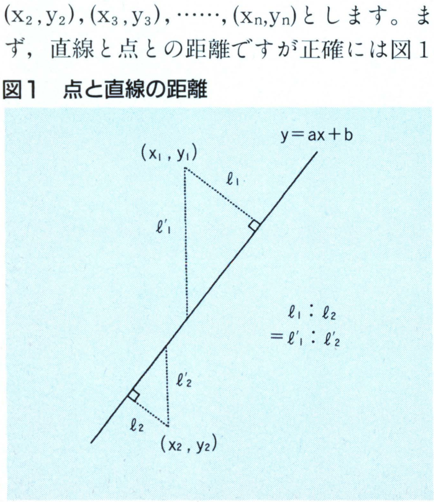
最小２乗法とは，１次式で表せる，または近似できる事象からその１次式を求める方法のひとつで，理屈では「すべてのデータとの差の２乗の和が最小になるような１次式の求め方」となります。なぜ２乗かというと，２乗すれば突出したデータがなくなり，誤差などが緩和できるからです。３乗，４乗すればもっといいのですが，それにかかる時間，手間に対して精度があまり上がらないという問題があります。

以上，だいたいこんなもんだと思ってもらえれば大丈夫です。で，どうやって求めるか考えていきましょう。ところで最近のポケコンにはこの機能が付いているものもあり，データをぽんぽんを入れるだけで求まってしまいますが，悲しいかな，ボンビーな私はポケコンが買えない。「それなら作ってやるわい！」というわけです。

解いてみる

まず，式を立ててみましょう。求めたい１次式を$y=ax+b$とし，データを(x_1, y_1)，(x_2, y_2)，(x_3, y_3)，……，(x_n, y_n)とします。まず，直線と点との距離ですが正確には図１

図1　点と直線の距離



でのl_1やl_2が正しい距離です。しかし，相手が直線であれば，垂直でない限りl_1とl_2の比はl_1'とl_2'の比に等しいことになります。最小２乗法ではとりあえずそれぞれのデータとの距離の比が正しければいいので，これらのｙ軸方向の差を使うことにします。$i=1, 2, 3, \cdots n$とすると，それぞれのデータと直線との差は(ax_i+b-y_i)で表されます。つまり，これらの２乗の和，

$$S=\sum_{i=0}^{n}(ax_i+b-y_i)^2$$

が最小となるようにa，bを求めればいいわけです。ではこの式を展開していきます。

$$S=\sum_{i=1}^{n}(ax_i+b-y_i)^2$$
$$=\sum_{i=1}^{n}(a^2x_i^2+2abx_i-2ax_iy_i+b^2-2by_i+y_i^2)$$
$$=a^2\sum_{i=1}^{n}x_i^2+2ab\sum_{i=1}^{n}x_i-2a\sum_{i=1}^{n}x_iy_i-b^2n-2b\sum_{i=1}^{n}y_i+\sum_{i=1}^{n}y_i^2 \quad \cdots\cdots①$$

ここで①をaで偏微分します。簡単にいうと，偏微分とはa以外はすべて定数として微分するようなものと考えてください。

$$\frac{\partial S}{\partial a}=2a\sum_{i=1}^{n}x_i^2+2b\sum_{i=1}^{n}x_i-2\sum_{i=1}^{n}x_iy_i$$
$$=a\sum_{i=1}^{n}x_i^2+b\sum_{i=1}^{n}x_i-\sum_{i=1}^{n}x_iy_i$$

①式で，a^2の係数$\sum_{i=1}^{n}x_i^2$は正数であるから，

$$\frac{\partial S}{\partial a}=0$$

とすればSが最小となるａの値が求まります。

$$a\sum_{i=1}^{n}x_i^2+b\sum_{i=1}^{n}x_i-\sum_{i=1}^{n}x_iy_i=0 \quad \cdots\cdots②$$

同様に②をｂで偏微分して，

$$\frac{\partial S}{\partial b}=2a\sum_{i=1}^{n}x_i+2bn-2\sum_{i=1}^{n}y_i$$
$$=a\sum_{i=1}^{n}x_i+bn-\sum_{i=1}^{n}y_i$$

①式で，y^2の係数ｎは正数だから，

$\frac{\partial S}{\partial b}=0$　とすると，

$$a\sum_{i=1}^{n}x_i+bn-\sum_{i=1}^{n}y_i=0 \quad \cdots\cdots③$$

これでa，bの関係式が２つ出たので，②③を連立して解きます。

③より，

$$b=\frac{\sum_{i=1}^{n}y_i-a\sum_{i=1}^{n}x_i}{n} \quad \cdots\cdots④$$

④を②に代入すると，

$$a\sum_{i=1}^{n}x_i^2+\frac{1}{n}\sum_{i=1}^{n}x_i\sum_{i=1}^{n}y_i-\frac{a}{n}\left(\sum_{i=1}^{n}x_i\right)^2-\sum_{i=1}^{n}x_iy_i=0$$

$$a\left\{\left(\sum_{i=1}^{n}x_i\right)^2-n\sum_{i=1}^{n}x_i^2\right\}=\sum_{i=1}^{n}x_i\sum_{i=1}^{n}y_i-n\sum_{i=1}^{n}x_iy_i$$

$$a=\frac{n\sum_{i=1}^{n}x_iy_i-\sum_{i=1}^{n}x_i\sum_{i=1}^{n}y_i}{n\sum_{i=1}^{n}x_i^2-\left(\sum_{i=1}^{n}x_i\right)^2} \quad \cdots\cdots⑤$$

⑤を④に代入すると，

$$b=\frac{\sum_{i=1}^{n}y_i\left\{n\sum_{i=1}^{n}x_i^2-\left(\sum_{i=1}^{n}x_i\right)^2\right\}}{n\left\{n\sum_{i=1}^{n}x_i^2-\left(\sum_{i=1}^{n}x_i\right)^2\right\}}-\frac{\sum_{i=1}^{n}x_i\left(n\sum_{i=1}^{n}x_iy_i-\sum_{i=1}^{n}x_i\sum_{i=1}^{n}y_i\right)}{n\left\{n\sum_{i=1}^{n}x_i^2-\left(\sum_{i=1}^{n}x_i\right)^2\right\}}$$

$$b=\frac{\sum_{i=1}^{n}x_i^2\sum_{i=1}^{n}y_i-\sum_{i=1}^{n}x_i\sum_{i=1}^{n}x_iy_i}{n\sum_{i=1}^{n}x_i^2-\left(\sum_{i=1}^{n}x_i\right)^2} \quad \cdots\cdots⑥$$

となります。

以上の式の展開を行って最終的に得られた⑤⑥式からわかるように，

1)　$\sum_{i=1}^{n}x_i^2$

2)　$\sum_{i=1}^{n}x_i$

3)　$\sum_{i=1}^{n}x_iy_i$

4)　$\sum_{i=1}^{n}y_i$

を得ればいいことになります。プログラムではそれぞれ，sx2，sx，sxy，syという変数名を使っています。また，１次式を求めるだけでは味気ないので，グラフ表示してみました。リスト１にはサンプルデータが

入っていますので，自分でデータを入力したいなら，先頭のデータ数～目盛り幅までを適当に入力してください。

A)　データ数

データの個数を変数nに与える。

B)　データ

x，yデータをそれぞれ配列変数x，yに与える。データ数によって配列のサイズの変更を忘れないように。

C)　表示範囲

グラフ表示するxy方向の範囲を与える。xy方向それぞれxmin～xmax・ymin～ymaxの範囲となる。

D)　目盛り幅

xy軸の1目盛りの大きさをそれぞれdx，dyに与える。必ず軸端に目盛りがくるように設定すること。要するにdxをxminとxmaxの公約数にする。dyについても同じ。

積分

それではもうひとつ，今度は積分について考えてみましょう。いま，

$$S=\int_a^b f(x)dx$$

という式があったとします。この式はa～bまでの間を等間隔に薄くスライスして，それぞれの薄板を長方形とみなして面積を計算しようということです。いや，面積というのは語弊があるかもしれません。マイナスの値を取ることもありますし，プラスとマイナスで打ち消し合ってしまいますから。とにかく，上の式は図2のように$a=x_0$，$b=x_n$としてa～bまでを等分したとき，次式で表されます。

$$S=\lim_{n\to\infty}\frac{b-a}{n}\sum_{k=0}^{n-1}f(x_k)$$

これをシミュレートしてみましょう。nを無限大にするのは不可能ですので，徐々に大きくしていき，ある程度の精度が出たら終了ということにしましょう。ここではnは2から4，8，16……と1ステップごとに倍にしていくことにします。また，n分割したときの値をS_nとしたとき，

$$\varepsilon_n=|S_n-S_{n/2}|$$

が指定した数値以下になったときに終了することにします。本来なら，nが大きくなるたびにε_nが0に近づくことを証明する必要があるのでしょうが（いや，ひょっとしたら証明できないかもしれない），たぶん大丈夫だろうという楽観的な考えにより省略しました。

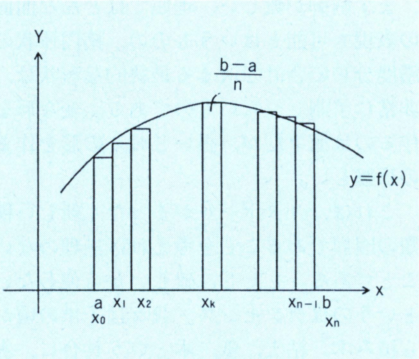
また，最終的な値は$S=S_n+\varepsilon_n$ということにしました。これも私のカンにしかすぎませんが，いくつかの式で試したところ，だいたい正解に近いものになりましたので，恐らく正しいのでしょう（どなたか証明してもらえませんか）。

リスト2にはリスト1と同様にサンプルデータが入っていますので，各自で計算する場合はプログラムを書き換えてください。先頭の積分範囲，終了条件，及び390行の被積分式です。

A)　積分範囲

lowerに下限をupperに上限を与える。

図2　積分



B)　終了条件

ε_nの最大値を与える。ε_nがこの値より小さくなったとき終了する。だいたいの値でいいときは0.1～0.01，精度を出したいときは0.001～0.0001程度。あまり小さな値を入れて30ステップを超えると，分割数がint型変数であるためにオーバーフローするので注意。

C)　被積分式

積分したい式を$y=f(x)$の形式で与える。こうすれば積分不可能な関数も近似値ですが積分できますし，計算結果の真偽を確かめることもできますからね。

以上，夏休みの課題開発支援記事みたいになってしまいました。リストは短いので適当なデータを作って遊んでみてください。

リスト2

```
 10 /*
 20 /* 積分
 30 /* 積分範囲・終了条件・披積分式を入力して実行
 40 /*
 50 /* 積分範囲
 60 float lower=0,upper=3.1415926535898#
 70 /* 終了条件
 80 float e=0.0001#
 90 /*
100 float s0,s1
110 int i,j
120 float l,dx
130 l=upper-lower
140 j=1
150 integral()
160 print
170 /* メインループ
180 while abs(s1-s0)>e
190  integral()
200  print "e=";s1-s0
210 endwhile
220 print "S=";2*s1-s0
230 end
240 /* 加算部分
250 func integral()
260  j=j*2
270  dx=l/j
280  print "dx=";dx,
290  s0=s1
300  s1=0
310  for i=0 to j-1
320   s1=s1+f(lower+dx*i)*dx
330  next
340  print "S";j;"=";s1,
350 endfunc
360 /* 披積分式
370 func float f(x;float)
380  float y
390  y=cos(x)
400  return(y)
410 endfunc
```

リスト1

```
 10 /*
 20 /* 最小二乗法
 30 /* データ数・データ・表示範囲・目盛りを入力して実行
 40 /*
 50 /* データ数
 60 int n=6
 70 /* データ    ↓ここの変更を忘れないように
 80 dim float x(6-1)={-2,-1,0,1,2,3}
 90 dim float y(6-1)={-3,-1,0,2,5,6}
100 /* 表示範囲
110 float xmin=-2,xmax=3,ymin=-3,ymax=6
120 /* 目盛り幅
130 float dx=0.5#,dy=0.5#
140 /* ここまで入力
150 float a,b
160 float sx,sy,sxy,sxsy,sx2,s2x
170 int x0,x1,y0,y1
180 int lx1,lx2,ly1,ly2
190 screen 2,0,1,1
200 console ,,0
210 /* 1目盛りが10の倍数ドットになるように
220 /* 適当に軸の始点・終点を決める
230 x1=700*dx/(xmax-xmin)/10*10
240 x0=(768-x1*(xmax-xmin)/dx)/2-xmin*x1/dx
250 y1=500*dy/(ymax-ymin)/10*10
260 y0=(512-y1*(ymax-ymin)/dy)/2-ymin*y1/dy
270 /* X軸
280 line(x0+xmin*x1/dx,512-y0,x0+xmax*x1/dx,512-y0,5)
290 for i=0 to (xmax-xmin)/dx
300  /* X軸目盛り
310  line(x0+i*x1+xmin*x1/dx,515-y0,x0+i*x1+xmin*x1/dx,509-y0,5)
320 next
330 /* Y軸
340 line(x0,512-y0-ymin*y1/dy,x0,512-y0-ymax*y1/dy,5)
350 for i=0 to (ymax-ymin)/dy
360  /* Y軸目盛り
370  line(x0-3,512-y0-i*y1-ymin*y1/dy,x0+3,512-y0-i*y1-ymin*y1/dy,5)
380 next
390 for i=0 to n-1
400  /* データプロット
410  pset(x(i)*x1/dx+x0+0.5#,512-y(i)*y1/dx-y0+0.5#,11)
420 next
430 /* シグマの計算
440 for i=0 to n-1
450  sx=sx+x(i)
460  sy=sy+y(i)
470  sxy=sxy+x(i)*y(i)
480  sx2=sx2+x(i)*x(i)
490 next
500 sxsy=sx*sy
510 s2x=sx*sx
520 a=(sxy-sxsy/n)/(sx2-s2x/n)
530 b=(sy-a*sx)/n
540 print "y=";a;"x+";b
550 /* 1次直線表示
560 lx1=xmin*x1/dx+x0+0.5#
570 lx2=xmax*x1/dx+x0+0.5#
580 ly1=512-(a*xmin+b)*y1/dy-y0+0.5#
590 ly2=512-(a*xmax+b)*y1/dy-y0+0.5#
600 line(lx1,ly1,lx2,ly2,8)
610 end
```

モデル化による演算高速化

疑似メタボールで遊ぶ

Tan Akihiko 丹 明彦

真面目に取り組むと，とてつもない計算量があるメタボール。そこで，高速化のためにアルゴリズムを変え，モデル化によって見えにくい要素を切り落とす。目指すは，レスポンスのよい2次元メタボールエディタだ。

曲線を気持ちよく操りたい。ドローツール「CANVAS PRO-68K」はベジェ曲線をうねうねと操る快感を教えてくれた。内部ではこ難しい数式を使って計算されている曲線なのだが，そんなことは気にしない。制御点をつかんで引きずり回すだけで，滑らかで気持ちのよい曲線が手に入る。

今回は「2次元メタボール」をでっちあげ，ベジェ曲線とは違う操作性をもった自由曲線を扱うためのアプローチを示そう。さらに今回は，第1目標として気持ちよい操作を目指した。

メタボールとは

曲線を3次元に持ち込むと，それは曲面と呼ばれる。しかしまだ曲面を気持ちよく操れるツールにはお目にかかったことがない。それどころか自由曲面を扱えるソフトそのものが少ない。

自由曲面といえそうなものは，レイトレーシングソフト「C-TRACE＋」に搭載されたメタボール。変形する球，という名前の由来をもつメタボールは，互いに近づけると変形するという特殊な性質をもっている。

メタボールの原理は面白い。中心部が濃く周辺部が薄いという濃度分布をもったメタボールを空間に複数配置する。それぞれのメタボールの濃度を合計した濃度が，あらかじめ定められたしきい値を超えている部分が目に見える。

たとえば2つのメタボールを近づけると，近づいている部分が変形・融合し，ヒョウタン形になる。理由は，メタボールの周囲に見えないけれども濃度分布が存在していて，別のメタボールが近づいたときに濃度の合計がしきい値を超えるためである。

さらに，メタボールの中には「正のメタボール」と「負のメタボール」が存在する。「負」とは「正の数・負の数」の「負」，マイナスのこと。通常(正)のメタボールのそばに負のメタボールを置くと，正のメタボールはくぼんだり穴があいたりする。もともと正のメタボールがもっていた濃度を負のメタボールが打ち消し，その結果濃度の合計がしきい値を下回るためである。

正・負のメタボールを組み合わせると，多彩な造形が可能になる。どちらかといえば，乗用車などのもっている幾何学的な曲面よりも，生物のもっているようないわゆる「有機的な」曲面を得意とする。

気持ちよさとはなにか

メタボールにも欠点はある。

まず制御が難しい。理屈ではどんな曲面の表現も可能とはいうものの，楕円体状の濃度分布の合計で決まる最終的な形状は，非常に予測しづらいものである。変な形を作るのは簡単だが，狙いどおりの形を作るのは難しい。

これは，メタボールがまったく新しい種類の図形であることを考えれば無理のないことである。しかし，難しいから使わないというのはナンセンス。我々は子供の頃から積み木，粘土，紙，木という具合に，次次と新しい材料を手に入れて造形法をマスターしてきた。メタボールもその延長にすぎない。

いま，メタボールという新しい材料を手に入れたのだから，制御の仕方を体得すればいいだけのこと。試行錯誤してコツを身に着ける以外に道はない。

だが，その試行錯誤を妨げるものがある。それは計算の重さだ。ちょっといじるたびに何分も待たされていたのでは，やり直しをする気力を失ってしまうものだ。

とりわけメタボールのように自由自在の変形をするものを気持ちよく操作するということは，ちょっとした操作の結果をすぐに見られるようなレスポンスのよいモデリング環境でのみ実現されるのである。

いかにサボるか

レスポンスのよい，ということは結局のところ，処理が速い，ということと等価である。処理速度を上げるためにはいくつかの戦略がある。

まず，そもそも速い計算機を使うという「力は正義」的なやり方。これはシリーズを通じて，ほとんど単一アーキテクチャのX68000においては意味のない話である。

ハードウェアが遅いならソフトウェアでカバーする。これなら安上がりだし，将来ハードが速くなったときには，ものすごく速い処理が実現することも期待できる(そういう意味で安易な高速ハードの投入は慎むべきだとも思う)。

ソフトウェアによる高速化を，ここでは2つに分類したい。ひとつはアルゴリズムによる高速化，もうひとつはコーディングによる高速化。

僕はアルゴリズムで速くするほうが大切だと思っている。アルゴリズムを工夫せずにいくら最適なコードを組んでも真に速くはならないことが多い。速いアルゴリズムを見つけて初めて最適化の努力をするというのが効率的である。

典型的な例でいえば高速フーリエ変換(FFT)。フーリエ変換の性質を上手に利用して劇的に速くするアルゴリズムである。これが真面目なフーリエ変換だと，たとえ理論上は最適なコードを組んだところで，FFTのパフォーマンスにはとうていかなわない。

今回は，2次元の疑似メタボールをプログラムしたわけだが，真面目に計算することは最初からあきらめ，少なくとも見かけは同等の処理をできるだけ効率よく実現することを目標とする。ある程度精度を犠牲にし，快適とはいえないまでもそこそこのレスポンスを実現したつもりだ。

2次元メタボール

メタボールというと3次元を意味するので，メタサークルなどという造語をしようかとも思ったが，「2次元メタボール」と頭に2次元をつけることでメタボールのまま通すことにした。

2次元メタボールの概念を図1に示す。通常(3次元)のメタボールは，これをそのまま3次元に拡張したものと思えばいいが，2次元のほうがイメージがつかみやすい。

メタボールを決めるパラメータは，

・中心座標　(x, y)
・半径　r
・符号を含めた中心部の濃度　Umax
・しきい値　t

などである。xy平面上に2次元メタボールを配置し，z軸方向に濃度(u)を取って3次元のグラフを描く。正のメタボールなら濃度分布は山のような形になり，負のメタボールなら穴ができる。

平面u=tで切った断面がメタボールの形となる(図2)。正のメタボールどうしが融合してヒョウタン形になったり，正と負のメタボールが相殺して三日月形になる様子がわかる。

これはあくまで概念。実際に図形としてメタボールを描こうとする場合，

(濃度の総和)=(しきい値)

なる方程式を立てて解く必要がある。この方程式の解は閉曲線となる(3次元の場合は閉曲面)。レイトレーシングで使われているメタボールの場合，この曲面と視線との交点を求めようというのだから，けっこう計算量は多い。ちなみに，レイトレーシングのプリミティブである球と直線の交点のように，2次方程式の解の公式が適用できる問題ではない。

モデル化

ほしいのはレスポンスである。とりあえず精度は必要ではない。今回は方程式を解くというアプローチをあっさり放棄し，代案として，濃度分布を調べる領域を全部配列にもつという物量作戦に出ることにした。画面に表示するのだから，配列の要素数は画面のドット数(512×512)もあればいい。表示は256色モードだ。記録する濃度も16ビット整数で十分。もう割り切りまくっている。

配列は用意した。ここにメタボールを直接書き込むのだ。正のメタボールなら濃度の値を足し，負のメタボールなら値を引く。その計算結果を常に画面に反映させていれば，そこそこレスポンスのいい2次元メタエディタの出来上がりだ。もちろん，画面にアクセスするのは値の更新された部分だけだ。いちいち全画面を描き換えるなんて真似はしない。

さて，配列に書き込めばいいことはわかった。次はどう書き込むかだ。

正直にやるなら，中心座標と半径から濃度を求めて1ドットずつ書き込む。よさそうに思える。だがだめだ。なぜか。計算量が多いのだ。

図1と図2からもわかるのだが，ひとつのメタボールの影響する範囲はその中心から半径rの円内のみである。濃度分布は中心からの距離の関数。中心部をピークとしてなだらかな曲線を描き，境界部では0になっている。ひとつのメタボールの計算は，その勢力範囲の中だけで行えばよい。

ここまではいい。たとえば半径100ドットのメタボールを書き込むことを考えよう。勢力範囲内の各ドットについて，中心からの距離を求め(平方根の計算が必要)，それを使って濃度を計算し，値を足す。これを100×100×3.14=31400回も繰り返すのか？　だめだだめだ。

物量作戦その2。濃度分布も配列にもつのだ。先ほどもいったように，メタボールの勢力範囲は限られている。それを包み込

図1　2次元メタボール

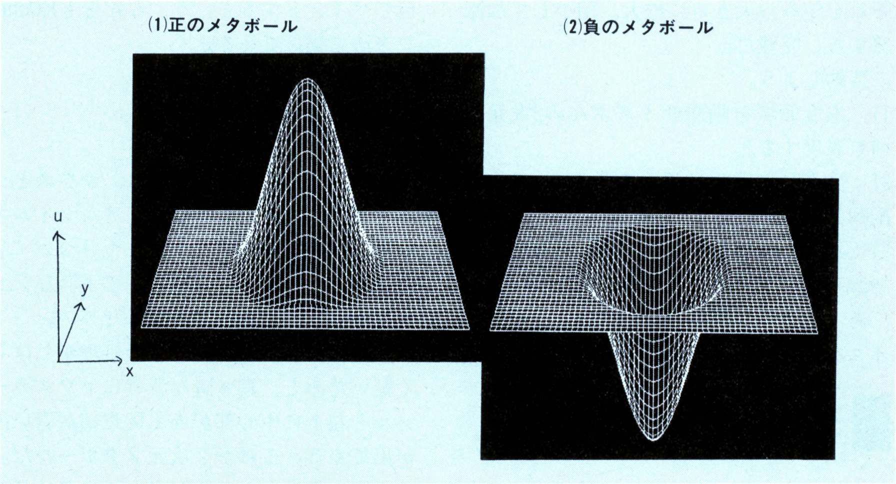
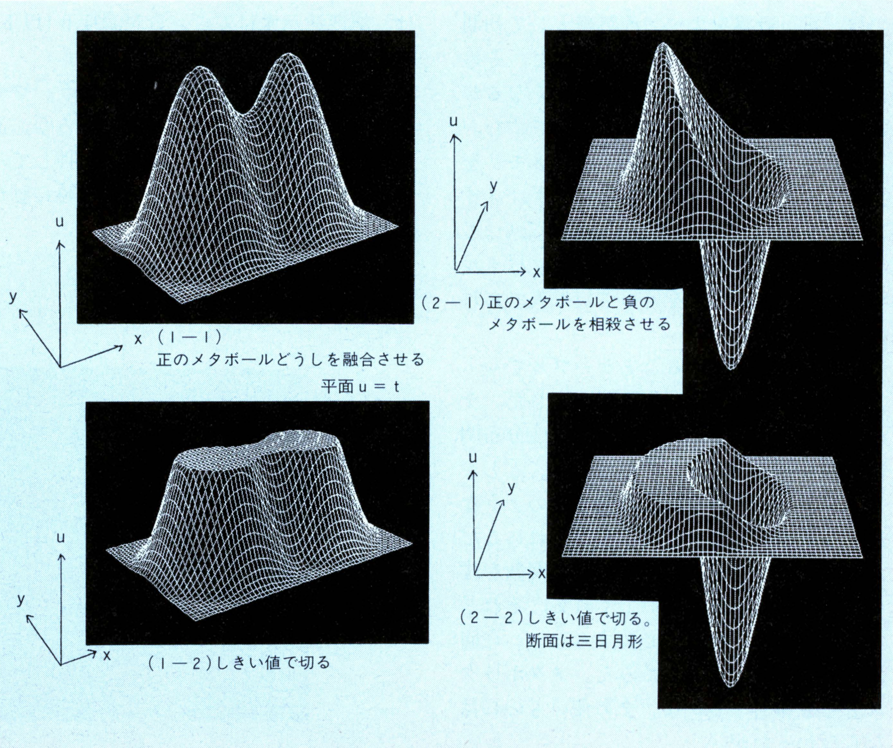


図2　メタボールを干渉させてしきい値で切断する



む長方形の中の濃度分布をまるごとテーブルに持つ。2次元配列にすることで，中心からの距離を求める必要はなくなる。話はもはや単純ループ＋加減算に落ちた。

でもメタボールの半径は一定ではない。半径200のもあれば半径20のもある。その全部に対してテーブルをもつのか？　いくら物量作戦万歳といっても，非現実的すぎはしないか？

ここで拡大，縮小を使うのだ。頭は使いよう。リアルタイム拡大，縮小は，アクションゲームだけの技術じゃない。ほしいのがレスポンスであれば，ゲーム向けの技術も使おうではないか。

適当な大きさの，そう256×256くらいの濃度分布テーブルをひとつ用意しておき，それを望みの大きさに拡大，縮小して加減算する。完璧だ。

要約しよう。

1)　濃度領域を画面サイズぶんの2次元配列で表現する。

2)　濃度分布関数は適当な大きさの2次元配列でもつ。

3)　拡大，縮小した(2)を(1)に書き込み，加減算の結果を画面に反映する。

あとはマウス関係のユーザーインタフェイスを入れて仕上げれば完成だ。

テーブル方式の副作用

今回はテーブルをしつこく活用することで速度向上を図っている。テーブルが速いのは，重い計算をすべて前処理として初期化ルーチンに押しつけているからだ。そのためプログラムの立ち上がりは遅くなるが，本処理はテーブルを参照するだけですむ。

そして，テーブル方式の欠点はメモリを食うこと。このプログラムにしても，メインメモリ1Mバイトではたぶん動かない。もっともgccでこのプログラムをコンパイルできるくらいの環境を持っている人であれば心配いらない。

濃度分布関数をまるごとテーブルでもったことにより，面白い副作用が生じた。テーブル化によって，いろいろな濃度分布関数を使うことができるのだ(図3)。もともとメタボールの分布関数は釣り鐘のような形である。でもプログラム上はそれにこだわることはなくて，円錐形の分布関数を作ってもプログラムは動いてくれる。それどころか，円にこだわる必要すらない。今回は冗談で四角形も入れてみた。メタボックスとでも命名するかな。まあ使いものにはならなかったが。

濃度分布関数テーブルを作成する手続きは関数makeU()の中にあるのだが，その正体は256色モードで画面に描いた図形をグラフィック取り込みを行うget( )関数で取り込んでいるというだけのものである。だから，濃度分布関数を描く手続きを差し換えれば，丸だろうが四角だろうが，取り込み画像だろうがOKなのである。

メタボールらしく使えそうな濃度分布関数は，プログラムリスト中でコメントアウトしていない三角関数を使ったものである。本来ならメタボールの濃度分布関数は，正式なもの(区間定義された多項式)があるが，三角関数でも十分いい振る舞いをしているのでよしとした。外側から半径を変えながら円を描き，その半径での濃度に対応する色でペイントするという，なんとも原始的な方法で描いている。

使い方

このプログラムを使うには，マウスとジョイスティックが必要だ。リアルタイムキー入力が面倒だったもんで，ついつい“<stick.h>”をインクルードして手軽なジョイスティックに走ってしまった……。

まず起動する。初期設定の処理がしばらく続いたあと，真っ青な背景にマウスカーソルと左上に中心部が赤く周辺部が青い円が出てくる。これが２次元メタボールだ。

画面右手には，色相が青から赤にカラフルに変化する帯が見える。このカラーバーは，濃度を示す目安で，青が濃度０(以下)，赤が濃度255(以上)を表している。

マウスカーソルを動かしてみよう。やや遅れてメタボールがついてくるだろう。適当なところでマウスの左ボタンを押してみよう。そのあとマウスカーソルを脇に動かすと，ボタンを押したところにメタボールが残っているはずだ。その近くにマウスカーソルを移動して，もう一度マウスの左ボタンを押してみよう。マウスカーソルを脇にどけると，２つのメタボールが重なり合い，中心部の赤い部分がヒョウタン形になっているはずだ。

今度は画面左上へマウスカーソルを持っていってマウス右ボタンを押してみよう。メタボールの大きさが変わる。そのあとマウスカーソルを動かすと，大きさの変わったメタボールがマウスカーソルについてくることだろう。

ジョイスティックのAボタンを押すと，メタボールは一転して負のメタボールになる。メタボールが消えたように見えるはずだ。その状態で，マウスカーソルをそれまでにメタボールを置いたところの近くに持っていくと，その部分の色が青っぽくなる。これが負のメタボールの効果だ。そこでマウス左ボタンを押すと，そこに負のメタボールが固定される。もう一度ジョイスティックのAボタンを押すと，正のメタボールに戻る。

適当にマウスカーソルを動かしてボタンを押していれば，だいたいの使い方はわかる。一応，以下にリファレンスを示す。特にジョイスティックの使い方はリファレンスがないとわからないだろう。

●マウスカーソル移動

メタボールを移動する。

●マウス左ボタン

メタボールを固定する。

●マウス右ボタン

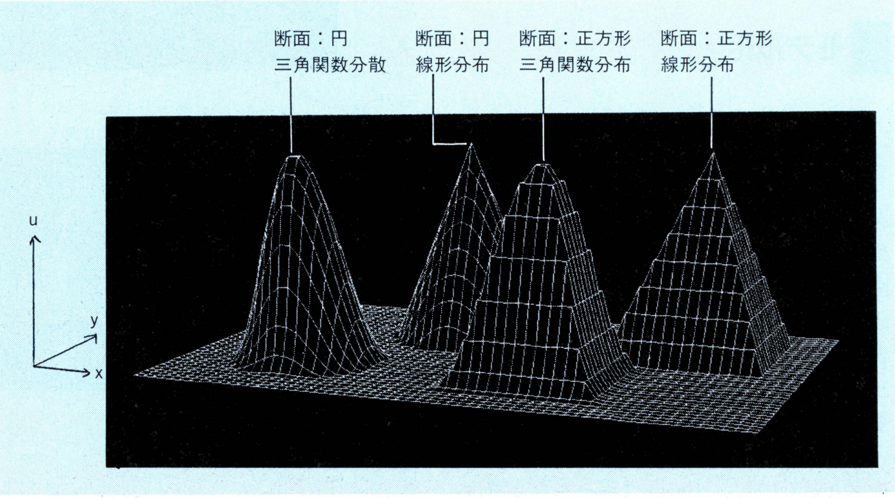
メタボールの大きさを変える。

●マウス右左ボタン

終了する。

●ジョイスティックAボタン

図3　さまざまな濃度分布



メタボールの正負を反転する。

●ジョイスティックBボタン

しきい値表示モードを切り替える。

●ジョイスティック上下

しきい値を増減する。

●ジョイスティック左右

しきい値表示の幅を増減する。

しきい値表示モードというのは，メタボールの最終出力をコントロールするのに使う。3モードあって，Bボタンを押すごとにサイクリックに切り替わる。

●モード0

通常のカラーバー。メタボールも青から赤の色相グラデーションのみの表示になる。

●モード1

カラーバーおよびメタボールの濃度分布のしきい値付近が白くなる。ほかはモード0と同じ。

●モード2

カラーバーおよびメタボールの，しきい値未満の部分は黒に，しきい値以上の部分は白になる。

このうち，モード2の状態がメタボールの最終出力(実際にレンダリングしたときの形)ということになる。要するにモード0〜2といっても，パレットを変えているだけのことだ。

ジョイスティック上下でしきい値を増減するというのは簡単に理解できることと思う。しきい値表示幅というのは，少しわかりにくいかもしれない。これは，画面の解像度が粗いとしきい値に対応するパレットだけを白に変えても，ちゃんと輪郭が表示されない恐れがあるのだ。つまり，パレットを変える範囲を広げられるようにした，というだけのことである。

正確を期すのであれば，パレット変更などという姑息な手段に頼らずに，ちゃんとしきい値による輪郭線を抽出することが必要なのだが，例によってレスポンスへの要求からこのような形に落ち着いている。

これが今回のプログラムの使い方のすべてだ。ひとつ問題を出そう。これを使ってドーナツ形を描いてみてほしい。正負1個ずつのメタボールで可能だ。

プログラム解説

コメントを豊富に入れたので，リストを読んだほうが早いが，キモの部分を少々。

プログラムの実行中，メタボールはマウスカーソルに追従して動くが，あれは毎回仮想空間(画面と同じ大きさで濃度分布を格納する作業領域)に書き込み，消している。それをある程度の速度で動作させるために，テーブルを多用している。テーブルを利用する大物としては，仮想空間とメタボール濃度分布関数がある。さらにいくつか「こんなところにまで……」といいたくなるようなテーブルの使い方をしている。

ひとつは，リアルタイム拡大，縮小のためのテーブル(関数makeResizeTable( )で作成)。256×256ドットの濃度関数をw×hドットに拡大，縮小するのに毎回Bresenhamのアルゴリズムを使っていたのでは，いかに整数だけを使う速いアルゴリズムだとはいえ，処理速度の足を引っ張る。

ここでは，wとhがともに画面サイズ，つまり512を超えないことを根拠に，Bresenhamアルゴリズムの計算結果を格納するテーブルを用意している。そして，拡大，縮小をやっている部分が単純な2重ループとテーブル参照になっていることに注目してほしい(関数resizeAndAdd( )，resizeAndSub( ))。

もうひとつは仮想空間の内容を読み出して表示するときに参照するテーブル(関数makeJudge( )で作成)。仮想空間での濃度の値は，正のメタボールと負のメタボールが入り混じるので，必ずしも0〜255の範囲に収まらない。そこで，それを0または255にまるめ込むのだが，ここでif文を2つも使いたくないという理由だけでテーブルにした。表示のための2重ループのどまん中なので，そこそこに効果はあったと思う。

課題と展望

今回のプログラムの問題点。

・メタボールの濃度が変えられない

正負の反転はできても，強いメタボールと弱いメタボールが作れない。

・回転ができない

拡大，縮小はやっているが回転はしていない。

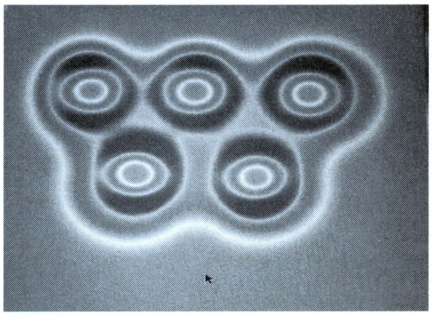
・大きなメタボールの反応が悪い

まだまだ処理が重い。X68000 XVIでも使えばかなり快適だが……。

これらの問題があっても，2次元ではそこそこのレスポンスを得られそうだ。アセンブラで書けば，完全リアルタイムも不可能ではあるまい。だが2次元では使い道があまりない。しいていえばドローツールの自由曲線としての道だろうか。

そして，このやり方で3次元化は可能だろうか。こうなるとほとんど「CPUパワーで勝負」の世界である。もちろん，いまの解像度で3次元化すれば，X68000のメモリでは扱いきれない(単純計算でも，256×256×256ですでに16Mバイト必要)。もっと解像度を落として，補間をうまく使うことで対処できるかもしれない。

正，負のメタボールを置き続けるとこんな感じ



ちょっと遊んでみました

終わりに

昔，C-TRACE＋をレビューしたとき，似たようなプログラムを作ったことがある。あのときはユーザーインタフェイスもなく，色使いも洗練されていなかったし，2次元メタボールを1枚描くだけでも数分を要していた。リアルタイムは無理だとあきらめていたのだ。

遅いシステムの開発者の多くは，この「無理だよ」という先入観で勝手に納得してしまっているのではないだろうか。

そうしたなかで，徹底して割り切って，レスポンスのいいシステムを作る。すばらしいブレイクスルー。このテのもので最近いちばん感動させられたのは，なんといってもPCM8だ。

最後にちょっとだけ夢，というか私がCG世界に感じていることを話す。それは，CGシステムが遠からず完全リアルタイムになるだろう，ということだ。もはや，モデリングとレンダリングの区別も存在しなくなるかもしれない。操作の結果が即座に画面に反映し，かつそれが目的の画像だったら……。真のバーチャルリアリティである。暴言かもしれないが，いまの「バーチャルリアリティ」は，「データグローブ」と同義語のような気がしてならないのだ。

リスト

```c
/*
 *	pmeta.c
 *	- 疑似メタボール(2次元)
 *
 *	コンパイル:
 *	    gcc -O -Wall meta.c -lbas -lfloatfnc
 *
 *		1992/07/19	丹 明彦
 */

#include	<basic0.h>
#include	<math.h>
#include	<graph.h>
#include	<mouse.h>
#include	<stick.h>

#define		UMAX	255			/* メタボール中心の濃度 */
#define		UMIN	0			/* メタボール周辺の濃度 */

unsigned char	u[256][256];			/* メタボールの濃度関数テーブル */
short		field[512][512];		/* メタボールの「場」(仮想空間) */
unsigned char	resizeTable[512][512];	/* リサイズ用のテーブル */

/*
 *	void setupPalet()
 *	- 濃度分布に対応したパレットを設定する(256色モード)
 *	   濃度0～255に対応する色は
 *	   青から水色・緑・黄色・橙 を経て赤までのグラデーション
 *	   (hsvの色相だけを変化させたもの)
 *	   しきい値に関して3つの表示モードがある
 */

int	paletMode = 0;		/* しきい値の表示モード */
int	threshold = UMAX/2;	/* しきい値 */
int	bandWidth = 2;		/* 表示に幅を持たせるための値 */

void	setupPalet()
{
	int	i;

	switch ( paletMode ) {
	case 0:			/* デフォルトのパレット */
		for ( i = UMIN; i <= UMAX; i++ )
			palet( i, hsv( 127-(i/2), 31, 31 ) );
		break;
	case 1:			/* しきい値付近を白にする */
		for ( i = UMIN; i <= UMAX; i++ )
			palet( i, hsv( 127-(i/2), 31, 31 ) );
		for ( i = -bandWidth; i <= bandWidth; i++ )
			palet( threshold+i, 65535 );
		break;
	case 2:			/* しきい値より上を白, 下を黒にする */
		for ( i = UMIN; i <= UMAX; i++ )
			palet( i, (i>=threshold)?65535:0 );
		break;
	}
	return;
}

/*
 *	void cyclePalet()
 *	- パレットを切り替える
 *	   3つのモードを順番に切り替える
 */

void	cyclePalet()
{
	paletMode++;
	paletMode %= 3;
	setupPalet();

	return;
}

/*
 *	void makeResizeTable()
 *	- メタボールを高速に計算するためのテーブル
 *	   resezeTable[wまたはh][i=0～(w-1)または(h-1)]を計算する。
 *	   256×256ドットのメタボール濃度関数テーブルを
 *	   w×hドットに拡大縮小するためのもの。
 *	   1から512までの値をとりうるwまたはhに対して
 *	   256の幅(または高さ)の濃度関数テーブルを高速に参照できる。
 *	   Bresenhamアルゴリズムを用いている。
 */

void	makeResizeTable()
{
	int	dx, x, y, e, dx2, dy2;

	/*screen ( 1, 3, 1, 1 );				デバッグ用*/
	for ( dx = 1; dx <= 512; dx++ ) {
		/*
		 *  (0,0)-(dx,255)の線分を発生する
		 *	Bresenhamアルゴリズム
		 */
		dy2 = 255*2;
		dx2 = dx*2;
		e = -dx;
		y = 0;
		for ( x = 0; x < dx; x++ ) {
			/*pset( x, y, 63 );	デバッグ用*/
			resizeTable[dx-1][x] = y;
			e += dy2;
			while ( e >= 0 ) {
				y++;
				e -= dx2;
			}
		}
	}
	return;
}

/*
 *	void makeU()
 *	- メタボール濃度関数テーブル u[256][256]を生成する
 *	   画面に256×256ドットの濃度分布を描いて取り込んでいるので
 *	   円形に限らずどんな形でもよい。
 *	   この関数は前処理なのでどんな形でも本処理の実行時間は同じ。
 *	   濃度の範囲は各点で0～255。
 */

void	makeU()
{
	int	r, c;

	for ( r = 0; r < 128; r++ ) {
		/* 線形分布(あまりきれいに変形しない)
		circle( 127, 127, 128-r, r*2, 0, 360, 256 );
		paint( 127, 127, r*2 );*/

		/* 三角関数分布(きれいに変形する) */
		c = UMAX/2 + (int)(UMAX/2)*(cos((r-128)*3.14/128.0));
		circle( 127, 127, 128-r, c, 0, 360, 256 );
		paint( 127, 127, c );

		/* メタ矩形(ジョーク): 線形分布
		fill( r, r, 255-r, 255-r, r*2 );*/

		/* メタ矩形: 三角関数分布
		c = UMAX/2 + (int)(UMAX/2)*(cos((r-128)*3.14/128.0));
		fill( r, r, 255-r, 255-r, c );*/
	}
	/* まるごと取り込んで濃度関数テーブルとして使う */
	get( 0, 0, 255, 255, u, 256*256*sizeof(char) );

	return;
}

/*
 *	void initField()
 *	- メタボール濃度場(仮想空間)を初期化する
 *	   全領域の濃度を0にするだけ
 *	   (おまけでカラーバーも描く)
 */

void	initField()
{
	int	i, j;

	for ( i = 0; i < 512; i++ )
		for ( j = 0; j < 512; j++ )
			field[i][j] = 0;
	for ( i = UMIN; i <= UMAX; i++ )
		for ( j = 502; j < 512; j++ )
			field[i][j] = UMAX - i;

	return;
}

/*
 *	void makeJudge()
 *	- 表示時の判定高速化テーブルを作る
 *	   メタボールは加減算を行うので計算の結果
 *	   濃度がUMAXを越えたりUMINを下回ったりする
 *	   表示できるのはUMIN(=0)～UMAX(=255)の範囲
 *	   それを描画前に補正する
 *	   それ以下はUMINに、それ以上はUMAXに丸める
 *	   ifを使うと遅くなるのでテーブルを作っておく
 */

unsigned char	judge[32768];

void	makeJudge()
{
	int	i;

	for ( i = -16384; i <  UMIN; i++ ) judge[16384+i] = 0;
	for ( i =   UMIN; i <= UMAX; i++ ) judge[16384+i] = i;
	for ( i = UMAX+1; i < 16384; i++ ) judge[16384+i] = 255;

	return;
}

/*
 *	void	putField( x, y, w, h )
 *	- (x,y)-(x+w,y+h)の領域の濃度場(仮想空間)の内容を
 *	   画面に出力する
 *	   makeJudge()関数で作成したテーブルを利用する
 */

void	putField( x, y, w, h )
int	x, y, w, h;
{
	int			i, j;
	static unsigned char	ubuf[512];

	for ( i = 0; i < h; i++ ) {
		for ( j = 0; j < w; j++ ) {
			ubuf[j] = judge[ 16384 + field[y+i][x+j] ];
		}
		put( x, y+i, x+w-1, y+i, ubuf, w );
	}
	return;
}

int	xp = 0, yp = 0;		/* 直前に描いたメタボールの座標 */
int	wp = 128, hp = 128;	/* 直前に描いたメタボールのサイズ */
int	sp = 1;			/* 直前に描いたメタボールの正負 */

/*
 *	void preserve( x, y, w, h, s )
 *	- 直前に描いたメタボールの座標・サイズ・符号を保存する
 */

void	preserve( x, y, w, h, s )
int	x, y, w, h, s;
{
	xp = x;
	yp = y;
	wp = w;
	hp = h;
	sp = s;

	return;
}

/*
 *	void resizeAndAdd( x, y, w, h )
 *	- 濃度場(仮想空間)の領域(x,y)-(x+w,y+h)に
 *	   w×hドットに拡大縮小したメタボール濃度を加算する
 *	   正メタボールを置くときや負メタボールを消すときに使う
```

▶以前はまったく読んでいなかった「ハードウェア工作入門」と「マシン語カクテルin Z80's Bar」。最近，内容が情報処理技術者試験の受験者にとって非常に嬉しいものなので，毎月楽しみにしています。　鈴木　政宏(19)宮城県

```
 */

void    resizeAndAdd( x, y, w, h )
int     x, y, w, h;
{
        int     i, j, i1;

        for ( i = 0; i < h; i++ ) {
                i1 = resizeTable[h-1][i];       /* テーブルを参照して拡大縮小 */
                for ( j = 0; j < w; j++ )
                        field[y+i][x+j] += u[i1][ resizeTable[w-1][j] ];
        }
        return;
}

/*
 *      void resizeAndSub( x, y, w, h )
 *      - 濃度場(仮想空間)の領域(x,y)-(x+w,y+h)から
 *        w×hドットに拡大縮小したメタボール濃度を減算する
 *        負メタボールを置くときや正メタボールを消すときに使う
 */

void    resizeAndSub( x, y, w, h )
int     x, y, w, h;
{
        int     i, j, i1;

        for ( i = 0; i < h; i++ ) {
                i1 = resizeTable[h-1][i];       /* テーブルを参照して拡大縮小 */
                for ( j = 0; j < w; j++ )
                        field[y+i][x+j] -= u[i1][ resizeTable[w-1][j] ];
        }
        return;
}

/*
 *      void resizeAndDraw( x, y, w, h, s )
 *      - 濃度場(仮想空間)の領域(x,y)-(x+w,y+h)に
 *        w×hドットに拡大縮小したメタボールを描き込む
 */

void    resizeAndDraw( x, y, w, h, s )
int     x, y, w, h, s;
{
        if ( s > 0 )
                resizeAndAdd( x, y, w, h );
        else
                resizeAndSub( x, y, w, h );

        return;
}

/*
 *      void resizeAndErase( x, y, w, h, s )
 *      - 濃度場(仮想空間)の領域(x,y)-(x+w,y+h)から
 *        w×hドットに拡大縮小したメタボールを消す
 */

void    resizeAndErase( x, y, w, h, s )
int     x, y, w, h, s;
{
        if ( s > 0 )
                resizeAndSub( x, y, w, h );
        else
                resizeAndAdd( x, y, w, h );

        return;
}

/*
 *      メインルーチン
 */

void    main()
{
        int     x, y, w, h, s, dx, dy;
        int     mx, my, mdx, mdy, bl, br;
        int     mxp, myp;

        /* 初期設定 */
        screen( 1, 2, 1, 1 );                   /* 512×512ドット256色モード */
        setupPalet();                           /* 色相グラデーションのパレット設定 */

        makeResizeTable();                      /* 拡大縮小用のテーブルを作る */
        makeU();                                /* 濃度分布関数テーブルを作る */
        initField();                            /* 濃度場(仮想空間)初期化 */
        makeJudge();                            /* 表示時の判定高速化テーブルを作る */

        mxp = 128; myp = 128;                   /* 初期マウス座標 */
        w = 128; h = 128;                       /* メタボールの初期サイズ */
        s = 1;                                  /* 最初は正メタボール */
        preserve( mxp-w, myp-h, w, h, s );              /* 初期値保存 */
        resizeAndAdd( 0, 0, 128, 128 ); /* 最初のメタボールを置く */

        mouse( 0 );                             /* マウス初期化 */
        mouse( 4 );                             /* ソフトキーボードを殺す */
        setmspos( mxp, myp );                   /* マウス座標設定 */
        putField( 0, 0, 512, 512 );             /* 最初だけ全画面描き換え */
        mouse( 1 );                             /* マウスカーソル表示 */

        while ( 1 ) {
                msstat( &mdx, &mdy, &bl, &br );
                /* ボタンを両方押せば終了 */
                if ( br != 0 && bl != 0 ) break;
                if ( br != 0 ) {
                        /*
                         *   マウス右ボタン押下時の動作
                         *      右ボタンのドラッグでメタボールのサイズを決める
                         *      マウスの移動を検知するたびに
                         *      古いものを消して新しいものを描く
                         *      右ボタンを離したときの
                         *      原点(画面左上)とマウス座標でできる矩形領域が
                         *      メタボールのサイズになる
                         */
                        while ( br != 0 ) {
                                /* 古いメタボールを消す */
                                resizeAndErase( xp, yp, wp, hp, sp );
                                putField( xp, yp, wp, hp );
                                /*
                                 *   新しいマウス座標を得る
                                 *   (古いメタボールを消す時間が長いため)
                                 */
                                mspos( &mx, &my );
                                mxp = mx; myp = my;
                                w = mx;
                                h = my;
                                /* 新しいメタボールを描く */
                                preserve( 0, 0, w+1, h+1, s );
                                resizeAndDraw( 0, 0, w+1, h+1, s );
                                putField( 0, 0, w+1, h+1 );
                                /*
                                 *   右ボタンが離されるか
                                 *   移動を検知するかするまで待つ
                                 */
                                while ( 1 ) {
                                        msstat( &mdx, &mdy, &bl, &br );
                                        if ( br == 0 ) break;
                                        if ( mdx != 0 ) break;
                                        if ( mdy != 0 ) break;
                                }
                        }
                        continue;
                }
                /*
                 *   マウス右ボタン押下時以外の動作
                 *      メタボールはマウスの動きにつれて動く
                 *      マウスカーソルはメタボールの右下端に位置する
                 *      マウスの移動を検知するたびに
                 *      古いものを消して新しいものを描く
                 *      画面左と上からはみ出さないようにつぶす(手抜き)
                 *      左ボタンを押したときには消さない
                 *      原点(画面左上)とマウス座標でできる矩形領域が
                 *      メタボールのサイズになる
                 */
                mspos( &mx, &my );
                /*
                 *      ジョイスティックの操作
                 *         ボタンA  メタボールの正負反転
                 *         ボタンB  パレットモード切り替え
                 *         上下       しきい値上下
                 *         左右       しきい値表示幅変更
                 */
                if ( strig( 1 ) & 2 ) {
                        cyclePalet();
                        /* ボタンが離されるまで待つ */
                        while ( (strig( 1 ) & 2) != 0 );
                }
                switch ( stick ( 1 ) ) {
                case 8:
                        threshold++;
                        if ( threshold > UMAX ) threshold = UMAX;
                        setupPalet();
                        break;
                case 2:
                        threshold--;
                        if ( threshold < UMIN ) threshold = UMIN;
                        setupPalet();
                        break;
                case 4:
                        bandWidth--;
                        if ( bandWidth < 0 ) bandWidth = 0;
                        setupPalet();
                        break;
                case 6:
                        bandWidth++;
                        if ( bandWidth > 32 ) bandWidth = 32;
                        setupPalet();
                        break;
                }
                if ( bl != 0 ) {
                        /* 左ボタン */
                        beep();
                        while ( 1 ) {
                                /*
                                 *   ボタンが離されると次へ行く
                                 */
                                msstat( &mdx, &mdy, &bl, &br );
                                if ( bl == 0 ) break;
                                /*
                                 *   移動を検知しても次へ行く
                                 *   (コメントアウトしてあるが
                                 *   コメントをはずすと
                                 *   動かしながら置ける)
                                 */
                                /*
                                if ( mdx != 0 ) break;
                                if ( mdy != 0 ) break;
                                */
                        }
                } else if ( strig( 1 ) & 1 ) {
                        /* ジョイスティックAボタンはメタボールの符号反転 */
                        /* 古いメタボールを消す */
                        resizeAndErase( xp, yp, wp, hp, sp );
                        putField( xp, yp, wp, hp );
                        s = -s;
                        /* ボタンが離されるまで待つ */
                        while ( (strig( 1 ) & 1) != 0 );
                } else {
                        /* 移動がなかったときは描きかえない */
                        if ( mxp == mx && myp == my ) continue;
                        /* 古いメタボールを消す */
                        resizeAndErase( xp, yp, wp, hp, sp );
                        putField( xp, yp, wp, hp );
                }
                /*
                 *   新しいマウス座標を得る
                 *   (古いメタボールを消す時間が長いため)
                 */
                mspos( &mx, &my );
                mxp = mx; myp = my;
                x = mx - w;
                y = my - h;
                if ( x < 0 ) x = 0;
                if ( y < 0 ) y = 0;
                dx = mx - x;
                dy = my - y;
                /* 新しいメタボールを描く */
                preserve( x, y, dx+1, dy+1, s );
                resizeAndDraw( x, y, dx+1, dy+1, s );
                putField( x, y, dx+1, dy+1 );
        }
        return;
}
```

▶ 大学生活にも慣れた現在，私は単位の３原則を発見した。1)　崖っぷちでは止まらない。2)　テスト前はやわらかい。テスト後には悪魔がささやく。　角谷　光憲(18)愛知県

V70ボードの活用

AFPPを使った3Dグラフィック

Nakamori Akira 中森　章

3Dグラフィックを高速化するために，V70ボードのAFPPに装備されている行列演算命令を使ってみます。結果はX68000とのやりとりに多少問題が見られましたが，期待どおりの数値を見せてくれたようです。

V70アクセラレータ（以後V70ボードと呼びます）は，日本電気の32ビットマイクロプロセッサであるV70と，その浮動小数点コプロセッサであるAFPPを搭載しているボードです。7月号ではV70ボードの概要について報告しました。今回は，もう少し別の観点からV70ボードを使ってみたいと思います。

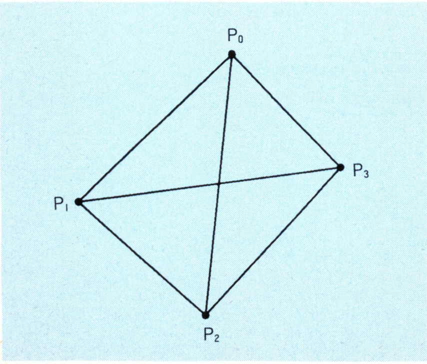
V70ボードの魅力は，CISCの集大成であるV70そのものにもありますが，なんといってもAFPPでしょう。浮動小数点の四則演算や三角関数などの計算ができるのは当然として，ベクトル・行列演算をサポートしていることが目新しいところです。ベクトル・行列といえば，コンピュータグラフィックスに関する計算でよくお目に掛かるデータ構造です。AFPPを使用すればこれらの計算を高速に行えるようになるに違いありません。AFPPはコンピュータグラフィックスがよく似合う浮動小数点コプロセッサなのです。

というわけで，今回はV70ボードを使用して簡単な3次元グラフィックを行ってみたいと思います。

3次元グラフィックの理論

3次元グラフィックの基本はワイヤーフレームモデルです。これは，起伏を示す線分（稜線）の組み合わせで物体を表現する方法です。創刊10周年記念PRO-68Kに付属した「SION II」やM.N.M Software（発売はビクター音楽産業）の「スターウォーズ」の画面を思い出してもらえば，説明は不要ですね。

図1　ワイヤーフレームモデルの例



ワイヤーフレームモデルのデータ構造は，頂点の集まりと頂点の繋ぎ方で決定されます。たとえば，図1のような三角錐は4つの頂点，

P_0，P_1，P_2，P_3

と6つの頂点の繋ぎ方（2つの頂点の組），

P_0-P_1，P_0-P_2，P_0-P_3，

P_1-P_2，P_1-P_3，P_2-P_3

によって決定されます。この図形が3次元の空間の中をどのように移動しようとも，どのように変形されようとも，頂点の繋ぎ方に変更はありません。

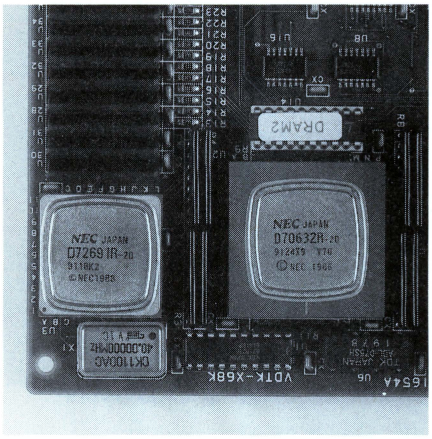
したがって，ワイヤーフレームモデルに対して移動・拡大・回転などの変換を加える場合は，それぞれの頂点がどのような位置になるかだけを注目していればいいことになります。要するに，ある点の位置が移動・拡大・回転によってどのように変化するかを知っておけば，ワイヤーフレームモデルの移動・拡大・回転は終わったも同然です。

詳しい説明は専門書に譲るとして，ここでは，紋切り的にある点を移動・拡大・回転する場合の位置変化を求める公式を挙げておきます。

なお，あとでAFPPのベクトル・行列命令を利用するために，形式がベクトルと行列を使って示してあります。また，以下の公式では元の点が(x, y, z)，変換後の点が(x′, y′, z′)になっています。

●X軸方向にdx，Y軸方向にdy，Z軸方向にdzだけ平行移動する場合

$$\begin{pmatrix} x' \\ y' \\ z' \end{pmatrix} = \begin{pmatrix} x \\ y \\ z \end{pmatrix} + \begin{pmatrix} dx \\ dy \\ dz \end{pmatrix}$$



これがV70とAFPP

●X軸周りに角度θだけ回転する場合

$$\begin{pmatrix} x' \\ y' \\ z' \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{pmatrix} \times \begin{pmatrix} x \\ y \\ z \end{pmatrix}$$

●Y軸周りに角度θだけ回転する場合

$$\begin{pmatrix} x' \\ y' \\ z' \end{pmatrix} = \begin{pmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{pmatrix} \times \begin{pmatrix} x \\ y \\ z \end{pmatrix}$$

●Z軸周りに角度θだけ回転する場合

$$\begin{pmatrix} x' \\ y' \\ z' \end{pmatrix} = \begin{pmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix} \times \begin{pmatrix} x \\ y \\ z \end{pmatrix}$$

●原点を中心にα倍に拡大する場合

$$\begin{pmatrix} x' \\ y' \\ z' \end{pmatrix} = \begin{pmatrix} \alpha & 0 & 0 \\ 0 & \alpha & 0 \\ 0 & 0 & \alpha \end{pmatrix} \times \begin{pmatrix} x \\ y \\ z \end{pmatrix}$$

次はワイヤーフレームモデルを画面に表示することを考えましょう。コンピュータの画面は2次元ですから，3次元の物体を2次元に変換することが必要になります。ここでは透視投影（中心投影法）という方法を使います。具体例としてZ軸上に視点を設け，原点付近にある物体を見る場合の公式を示しておきましょう。ここで，3次元空間の座標を(xy′z′)，2次元平面の座標を(x′,y′)とします。また，原点から視点までの距離をd1，視点から投影面の距離をd2と

します。

x′＝x×d2/(d1－z)

y′＝y×d2/(d1－z)

なぜこうなるのかは，図2で三角形の相似を考えれば明らかですね。

以上の公式を使い，図3に示すような手順で処理を行えば，ワイヤーフレームモデルを使った3次元グラフィックの出来上がりです。

データ構造と描画方法

3次元グラフィックの理論（というほど大袈裟なものではない）がわかったところで，これをどのようにプログラムするのか考えます。

3次元の点はdouble型を3つ組み合わせたものでよいでしょう。2次元の点はグラフィック画面に表示することを考えて，int型を2つ組み合わせたものにします。これらの点を配列で持てば頂点の集合の出来上がりです。頂点の集合が決まったら，次は頂点の組を示すデータ構造が必要になります。これは頂点を表す配列の添字で示せばよいでしょう。つまり，int型を2つ組み合わせたものになります。以上のデータ型をC言語のtypedefによって定義すると次のようになるでしょう。

```
typedef struct {
    double  x;
    double  y;
    double  z;
} POINT_3D;  /* 3次元の点 */

typedef struct {
    int     x;
    int     y;
} POINT_2D;  /* 2次元の点 */

typedef struct {
    long    fr_point;
    long    to_point;
} PT_PAIR;  /* 添字の組 */
```

このように，データ構造は単純なのですが，プログラムを書くうえで問題となるのは画面の描き換えです。3次元空間の中で物体を移動したり回転したりすると，頂点の座標は別の位置に移動してしまいますから，2次元平面に透視投影されている図形もそれに応じて描き直してやる必要があります。単純に考えれば，一度画面上のデータを消去してから新たな図形を描くことが考えられます。しかし，毎回画面を消去する方法では少なくとも2つの不都合があります。

ひとつ目は，画面を消してから描き直すまでの時間がかなりかかるので，画面がちらついて見にくくなるということです。1回の垂直表示期間で画面を消去する高速クリアを使ってもちらつきを抑えることはできません。2つ目の不都合は，表示されている物体が複数あるとき，ひとつの物体を描くたびに画面を消去していたのでは，画面上にはひとつの物体しか残らないことになります。すべての物体の移動や回転処理の終わるのを待って一度に描き直すのは，表示されている図形が切り替わるのに非常に時間がかかるため，現実的ではありません。

そこで，考えられるのは1本の線分（稜線）のみについて注目し，線分ごとに消しては描き直すという処理です。すなわち，ひとつの物体の移動や回転処理が終わったら，その物体を構成するすべての線分に関し，線分の単位で消しては描くという操作を繰り返します。こうすれば，どこかの線分が1本消えて，新しい線分が1本生じるだけですから，画面上の物体はどこかがちょっと変化したなという程度に感じられるだけです。この方法は画面がちらつくこともなく物体の動きの変化もスムーズです。

しかし，この方法は新たに描こうと思っている線分が，ひとつ前にどの位置にあったかを記憶しておく必要があります。線分を消すというのは，同じ位置にカラーコード0で線分を描き直すということです。そのために，2次元の頂点のデータを今回と前回の2種をもつことになりますが，ほかにいい案が思いつかなかったので（物体の数が少なければパレットを操作して消すという方法も考えられるが），今回はこの方法でプログラムを書いてみました。

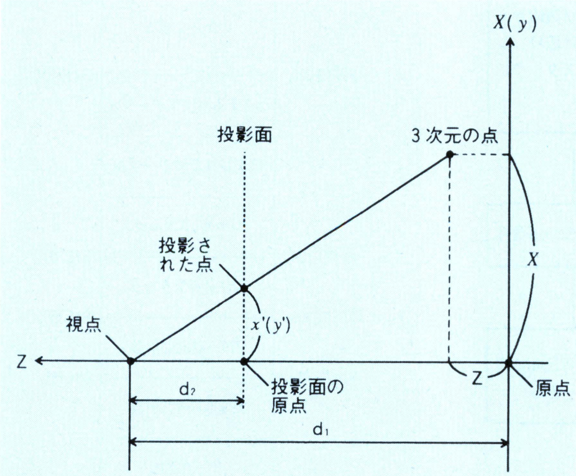
以上のような考えで書いたプログラムがリスト1です。まずはX68000上で動きを確認する目的ですべてC言語で記述してあります。Slideが物体を指定した距離だけX軸，Y軸，Z軸の3方向に平行移動させる関数，RotateByXが物体をX軸周りに指定した角度だけ回転させる関数，RotateByYが物体をY軸周りに指定した角度だけ回転させる関数，RotateByZが物体をZ軸の周りに指定した角度だけ回転させる関数，TransTo2Dが3次元の頂点を2次元の頂点に変換する関数，Displayが2次元に変換された頂点からなる図形をX68000の画面上表示（描画）する関数です。物体を示すためにOBJECTというデータ型を定義してありますが，これはこれまで説明してきたデータ型を組み合わせただけです。なお，物体の定義は6月号の付録である創刊10周年PRO-68Kのディスク4のSION IIのソースプログラムから自機，敵などの定義を持ってきたものです（座標は1/10になっています）。

ところで，リスト1のプログラムをコンパイルする場合は，とりあえずDEBUGというシンボルを定義してコンパイルしてください。つまり，GCCなら，

　　－DDEBUG

XCなら，

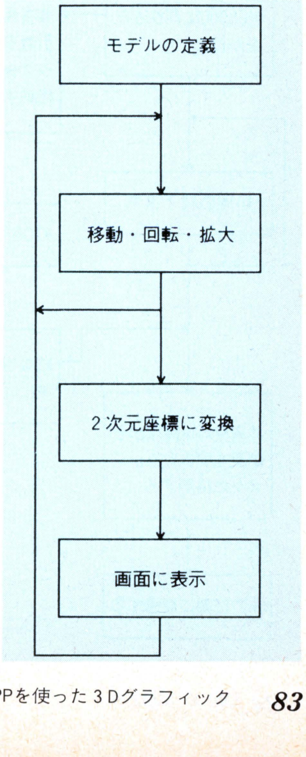
図2　透視投影



$$X' = X \times \frac{d_2}{d_1 - Z}$$

$$y' = y \times \frac{d_2}{d_1 - Z}$$

図3　3次元グラフィックの処理手順



/DDEBUG
というオプションを指定してコンパイルしてください。DEBUGというシンボルを定義しないと，あとに述べる高速版のプログラムになってしまいますからこのままでは動作しません。

リスト1のプログラムはV70ボードでも動作する予定だったのですが，付属のコンパイラ（VCC）が生成するコードを間違えるため，まともな図形は表示されません。どのような図形になるかはリスト1から，

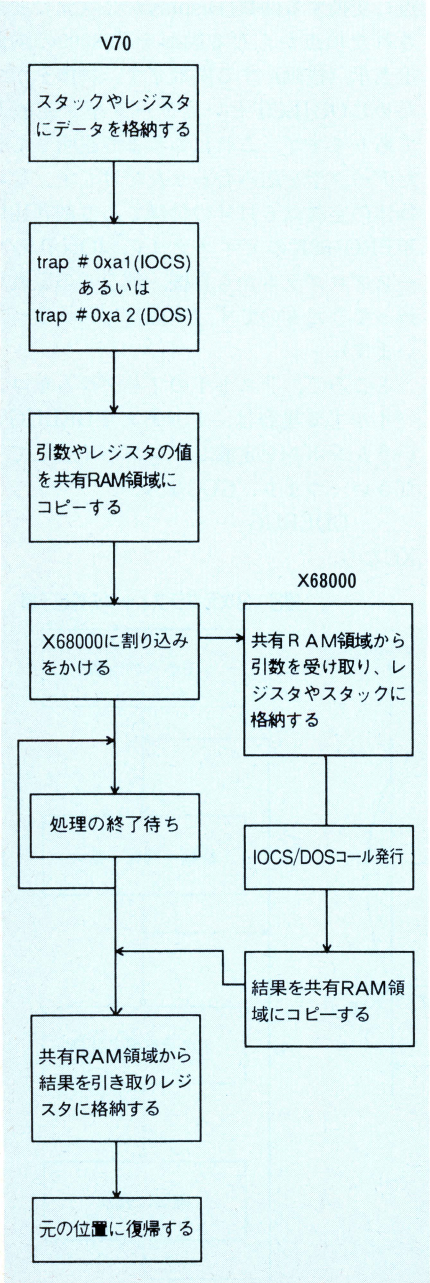
#ifndef V70

とそれに対応する，

#endif

を削除してVCCでコンパイルし動作させてみてください（VCCではV70というシンボルがあらかじめ定義されている）。コンパイラをうまく騙せばまともな図形を描くようにもできるのですが，空しいのでその方法の紹介はやめておきます。それよりも，誤動作する部分を含めてSlide，RotateByX，RotateByY，RotateByZなどの関数をアセンブリ言語で書き直して高速化を図ったほうが得策でしょう。

図4　IOCS/DOSコールの処理



高速化への試み

リスト1のプログラムをV70ボードで動作させるにあたり，AFPPの持つベクトル・行列演算を使用して高速化することを試みます。書き換え対象は3次元の座標変換を行うSlide，RotateByX，RotateByY，RotateByZという関数群です。AFPPのマニュアルをパラパラとめくっていると，3次元ベクトルに関する演算である，

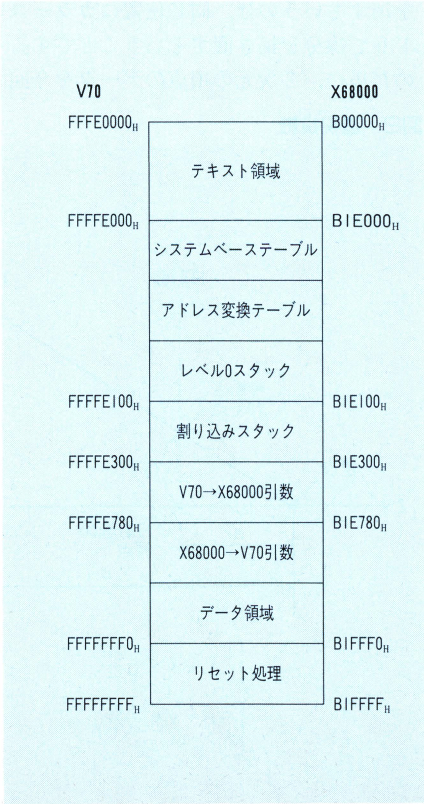
fmov3v.l（3次元ベクトルの移動）
fadd3v.l（3次元ベクトルの加算）
fmul3v.l（3×3行列と3次元ベクトルの積）

という命令が利用できそうなのがわかります。また，与えられた角度に対してsinとcosを同時に計算する，

sincos.l

という命令も便利そうです。これらを使ってSlide，RotateByX，RotateByY，RotateByZ関数を書き直したのがリスト2です。これらの関数のほかに，リスト2ではTransTo2D，Displayという関数も書き直してあります。これらはコンパイラのバグを回避するという意味もありますが，真の目的は高速化です。たぶん，コンパイラの出力する（であろう）コードよりも高速なものになっているはずです。

図5　SRAM（共有領域）のメモリマップ



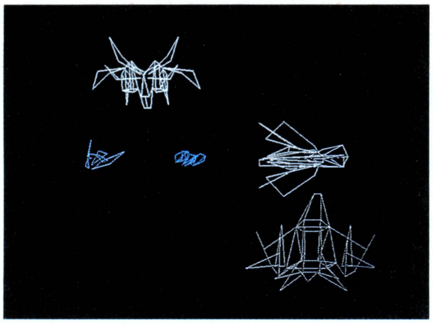
ところで，リスト2のプログラムの注釈を読めばわかりますが，かなり意味不明なことが書いてあります。これは第2の高速化を行った名残です。実は，リスト1の各関数をAFPP命令を使って書き直しても，実行速度はそれほど向上しません。それは，X68000側に理由があります。V70ボードではIOCSコールやDOSコールをX68000と通信しながら行っています。

たとえば，IOCSコールで直線を画面上に描く場合，実際にその処理を行っているのはX68000自身です。その場合の処理の概略は図4のようになります。図4を見てわかるように，引数を共有ＲＡＭ領域にコピーしたり，そこからデータを引き取ったりとメモリ操作が頻繁にあります。IOCSコールやDOSコールを行うために，V70側でトラップ(trap #0xa1，trap #0xa2)して，X68000側でもう一度トラップする（trap #15，FFxx$_H$）のもかなりのロスタイムです。つまり，V70やAFPPがいかに高速に動作しようともIOCSコールやDOSコールがあると，そのX68000側での処理時間が全体の実行時間の足を引っ張ってしまうのです。それを改善するための処置をリスト2のDisplayという関数では試みているのです。

まず，システム（V70IPL.V70）が引数の値をいちいち共有RAM領域にコピーし直すのは時間の無駄なので，プログラムの中から直接共有RAMに引数を書き込むようにします。そのためにプログラムを特権モードで実行する必要があるため，リスト2では特権モードに移行するための，

B_SUPER

という関数を定義してあります。X68000のスーパーバイザモードへ移行するライブラリ関数と同じ役割を持つ関数で，V70ボードで新たに作ってみました。そして，図5に私が独自に解析した，V70ボードのSRAM領域のメモリマップを載せておきます。引数の受け渡し領域はFFFFE300$_H$番地からのようですが，リスト2では安全を見込んでFFFFE400$_H$番地から引数を書き込んでいます。このとき，V70とX68000ではデータのバイト順序（エンディアンという）が



AFPPで高速化された3Dグラフィックのデモ

逆なので，バイト順序を変える処理も行っています。

次に，X68000側では共有RAM領域にある引数を直接受け取ることを考えます。このためにIOCSコールのベクタを書き換えて（フックする）専用のライン描画処理を行います。共有RAMから引数を勝手にコピーされるのを防ぐ（単に時間の無駄）ため，フックするIOCSコールは引数を持ってなく，ベクタを書き換えても影響が少ないものがよいと思われます。そこで選んだのがROMのバージョンを返すROMVER（ファンクション番号：$8F_H$）というシステムコールです。そのROMVERをライン描画に書き換えるためのプログラムがリスト3です。リスト3では共有RAM領域（V70ボードでの$FFFFE400_H$番地はX68000では$B1F400_H$番地に見える）を直接LINEPTR構造体の格納されている場所に指定して，IOCSコールであるLINE（ファンクション番号：$B8_H$）の本体をサブルーチンコールしているのです。ここでは，さらなる高速化のためにラインを消す処理と新たなラインを描く処理を同時に行っています。

リスト3ではIOCSコールでのライン描画ルーチンを直接コールしましたが，それを別のもっと高速なライン描画ルーチンに変更することも可能です。たとえば，村田敏幸氏の本（参考文献[2]）に紹介されているラインルーチンを使用すればかなり高速になります。

なお，リスト3はメモリ上に常駐するプログラムですが，処理を簡略化するために，2回常駐することを禁止する処理や常駐を解除する処理は入っていません。2回以上実行しないように注意してください。

最後になりましたが遊び方です。リスト1のプログラムが3d.cというファイル，リスト2のプログラムがsub.sというファイル，リスト3のプログラムがline.sというファイルに格納されているとします。X68000用の実行ファイルを作るために，XCでは，

cc /Y 3d.c

GCCでは，

gcc 3d.c −liocs −ldos

を実行します。これ以外の最適化オプションは適当に付けてください。これで3d.xという実行ファイルが生成されます。V70用の実行ファイルを作るためには，

vcc /Y 3d.c sub.s

を実行します。最適化オプションは付けないほうが無難でしょう。これでA.V70という実行ファイルが生成されます。同時に，

as line.s

lk line.o

によってline.xという実行ファイルを作っておいてください。以上のようにして，3d.x，A.V70，line.xができたら準備は終わりです。まず，

line

を実行してROMVERを書き換えておいてから，

3d

あるいは，

monv70 /i A

を実行します。前者がX68000による実行，後者がV70とAFPPによる実行です。処理時間が画面にプリントされますから，実行速度の違いを目で確かめてください。また，

3d 1

monv70 /i A 1

などのように，実行時にコマンドラインで何か引数を与えるとIOCSコール（ライン描画）をスキップした場合の処理時間が表示されます。これで，全体の処理時間のうちのどのくらいがライン描画に使われているかを知ることができます。

参考までに表1に処理時間の違いを示しておきます。ラインを描かない場合の速度を比較してみると，X68000側での処理の高速化がいかに大切かがわかると思います。

終わりに

私は3次元グラフィックが得意ではないので，今回のプログラムはかつてのOh!X（参考文献[1]）を参考にしながら書きました。3次元処理としては大したことはやっていませんが，V70ボードでAFPP命令を使うという目的に免じて許してくださいね。私としては，今回のプログラムはV70ボードでシステムモニタ（V70IPL.V70）やコマンドシェル（MONV70.X）を書き換えて遊ぶための第一歩と理解しています。与えられた使い方だけをしていたのでは面白くありませんからね。

今後，機会があればV70ボードの変則的な使用方法についてどんどん紹介していきたいと思います。

≪参考文献≫
1) 相馬英智，ワイヤーフレームによる3D世界，Oh!X，1988年9月号，pp.22-29.
2) 村田敏幸，X68000マシン語プログラミング グラフィックス編，ソフトバンク，1992年.
3) μPD72691ユーザーズ・マニュアル，IEM-5062C，日本電気，1989年.

表1　プログラムの実行速度

ライン処理	その1	その2	なし
V70	226.8	85.3	10.0
FLOAT2	560.9	419.5	378.3
VFLOAT	451.4	309.9	268.7

（単位：秒）

●初代X68000（10MHz）で測定。OPMDRVはOFF
●ライン処理その1はリスト3で示した処理をそのまま使用
●ライン処理その2は参考文献[2]のCHAPTER3，リスト14でクリッピングを省略した処理を使用

リスト1 3D.C

```
#include <iocslib.h>
#include <doslib.h>
#include <math.h>
/*
        データ型の定義
*/
typedef struct {
        double  x;
        double  y;
        double  z;
} POINT_3D;     /* 3次元座標 */

typedef struct {
        int     x;
        int     y;
} POINT_2D;     /* 2次元座標 */

typedef struct {
        long    fr_point;
        long    to_point;
} PT_PAIR;      /* 辺 */

typedef struct {
        int       np;     /* 頂点の個数 */
        POINT_3D *pt;     /* 頂点のデータ */
        int       nl;     /* 辺の個数 */
        PT_PAIR  *ln;     /* 辺のデータ */
        POINT_2D *wk1;    /* 2次元変換用ワーク */
        POINT_2D *wk2;    /* 2次元変換用ワーク */
        int      color;   /* 色 */
} OBJECT;
/*
        物体の定義
*/
#define N_PT1   41
#define N_LN1   51

POINT_3D _obj1[N_PT1]={ /* 物体の頂点の座標 */
{ 0.0, 1.0,-5.0},{ 1.5, 0.0, 1.5},{-1.5, 0.0, 1.5},{ 1.0, 0.0, 2.0},
{ 2.0, 1.0,-1.5},{ 0.5, 0.0,-4.5},{ 1.0, 1.5,-1.5},{-1.0, 0.0, 2.0},
{-2.0, 1.0,-1.5},{-0.5, 0.0,-4.5},{-1.0, 1.5,-1.5},{ 0.0, 0.0, 2.5},
{-1.0,-4.0, 4.5},{-0.5,-5.5,-6.0},{ 0.5,-5.5,-6.0},{ 1.0,-4.0, 4.5},
{-1.0,-1.0, 2.0},{-2.5,-3.5, 1.0},{-6.5,-1.0,-2.5},{-3.5, 1.5,-4.5},
{-0.5,-1.0, 2.5},{ 0.5,-1.0, 2.5},{ 1.0,-1.0, 2.0},{ 2.5,-3.5, 1.0},
```

```
{ 6.5,-1.0,-2.5},{ 3.5, 1.5,-4.5},{ 0.0, 1.0, 0.5},{ 0.0, 3.5, 1.5},
{ 0.0, 6.0, 4.0},{ 1.5, 2.0, 3.0},{ 2.0, 4.0, 6.0},{-1.5, 2.0, 3.0},
{-2.0, 4.0, 6.0},{-1.5, 1.5, 0.0},{-3.5, 2.5,-2.5},{-6.0, 5.5,-5.0},
{ 1.5, 1.5, 0.0},{ 3.5, 2.5,-2.5},{ 6.0, 5.5,-5.0},{ 0.0, 1.0, 3.0},
{ 0.0, 3.0, 6.5},
};

PT_PAIR _ln1[N_LN1]={    /* 辺を作る頂点の組 */
{ 0, 1},{ 2, 0},{ 3, 4},{ 4, 5},{ 5, 6},{ 6, 3},{ 6, 4},{ 7, 8},
{ 8, 9},{ 9,10},{10, 7},{10, 8},{11,12},{12,13},{13,14},{14,15},
{15,11},{16,17},{17,18},{18,19},{19,16},{ 2,20},{20,21},{21, 1},
{20, 0},{ 0,21},{12,15},{22,23},{23,24},{24,25},{25,22},{26,27},
{27,28},{26,29},{29,30},{29,27},{26,31},{31,32},{31,27},{28,30},
{32,28},{33,34},{34,35},{35,33},{36,37},{37,38},{38,36},{26,39},
{39,40},{40,30},{32,40},
};

#define N_PT2   51
#define N_LN2   75

POINT_3D _obj2[N_PT2]={
{-0.4, 2.0, 8.0},{ 0.4, 2.0, 8.0},{-1.0,-0.6, 3.0},{ 1.0,-0.6, 3.0},
{-2.0, 0.0, 2.0},{ 2.0, 0.0, 2.0},{-1.0,-1.0, 1.0},{ 1.0,-1.0, 1.0},
{-4.0, 1.0, 0.6},{-7.0, 2.6, 2.0},{-6.0, 2.0,-1.0},{-5.0, 1.4,-4.0},
{-4.0, 1.0,-6.0},{ 4.0, 1.0, 0.6},{ 7.0, 2.6, 2.0},{ 6.0, 2.0,-1.0},
{ 5.0, 1.4,-5.0},{ 4.0, 1.0,-6.0},{-3.0, 0.0,-5.0},{ 3.0, 0.0,-4.0},
{-1.0,-1.0,-4.0},{ 1.0,-1.0,-4.0},{-1.0,-1.0,-6.0},{ 1.0,-1.0,-6.0},
{-2.0,-1.0,-5.0},{ 2.0,-1.0,-5.0},{-4.0,-3.0,-8.0},{ 0.0,-2.0,-8.0},
{ 4.0,-3.0,-8.0},{-2.0, 0.0,-5.0},{-1.0, 0.0,-6.0},{ 1.0, 0.0,-6.0},
{ 2.0, 0.0,-5.0},{-1.0, 1.0,-2.0},{ 1.0, 1.0,-2.0},{-1.0, 0.0, 2.0},
{ 1.0, 0.0, 2.0},{-4.0, 2.0,-8.0},{ 4.0, 2.0,-8.0},{-4.0, 2.0, 3.0},
{-4.6, 1.6,-4.0},{-3.4, 1.6,-4.0},{ 4.0, 2.0, 3.0},{ 4.6, 1.6,-4.0},
{ 3.4, 1.6,-4.0},{-4.0, 1.0,-2.0},{-8.0, 0.0,-5.0},{-4.0, 1.0,-4.0},
{ 4.0, 1.0,-2.0},{ 8.0, 0.0,-5.0},{ 4.0, 1.0,-4.0},
};

PT_PAIR _ln2[N_LN2]={
{ 0, 1},{ 0, 2},{ 0, 4},{ 1, 3},{ 1, 5},{ 4, 2},{ 4, 6},{ 4,10},
{ 4,20},{ 4,18},{ 5, 3},{ 5, 7},{ 5,15},{ 5,21},{ 5,19},{ 2, 3},
{ 6, 7},{ 2, 6},{ 3, 7},{ 6,20},{ 7,21},{ 9,11},{11,12},{12,18},
{ 8,12},{14,16},{16,17},{17,19},{13,17},{18,20},{20,21},{21,19},
{18,24},{24,22},{22,23},{23,25},{20,24},{21,25},{25,19},{26,24},
{26,22},{27,22},{27,23},{28,25},{28,23},{29,18},{29,24},{32,19},
{32,25},{29,32},{33,29},{33,30},{33,35},{34,32},{34,31},{34,36},
{37,29},{37,30},{38,32},{38,31},{22,30},{23,31},{35, 4},{36, 5},
{35,36},{39,40},{40,41},{39,41},{42,43},{42,44},{43,44},{45,46},
{46,47},{48,49},{49,50},
};

#define N_PT3   56
#define N_LN3   68

POINT_3D _obj3[N_PT3]={
{-1.0, 0.0, 0.5},{ 1.0, 0.0, 0.5},{ 0.5,-1.0, 2.0},{-0.5,-1.0, 2.0},
{ 1.5,-1.5, 0.5},{ 1.0, 1.0, 2.0},{ 2.0, 1.0, 2.0},{ 3.0,-1.0, 0.5},
{ 0.0,-1.0, 2.0},{ 0.0, 1.5, 7.5},{ 3.0, 0.0,-2.5},{ 2.0, 0.0,-5.0},
{ 1.5,-0.5,-6.0},{ 2.5, 0.0,-4.0},{ 6.5, 2.5,-1.0},{ 7.5, 2.0, 3.0},
{ 5.0, 2.5,-0.5},{ 2.5, 0.5,-3.0},{ 2.0, 1.0,-1.5},{ 3.0, 2.0, 7.0},
{ 0.0,-1.5,-3.5},{-1.5,-0.5,-6.0},{ 0.5,-0.5,-2.0},{ 3.0,-3.0,-4.0},
{ 4.5,-3.5,-7.5},{ 3.5,-2.5,-3.5},{ 0.5, 0.5,-2.0},{ 3.5, 2.5,-3.5},
{ 4.5, 4.0,-7.5},{ 3.0, 3.0,-4.0},{-1.5,-1.5, 0.5},{-1.0, 1.0, 2.0},
{-2.0, 1.0, 2.0},{-3.0,-1.0, 0.5},{-3.0, 0.0,-2.5},{-2.0, 0.0,-5.0},
{-2.5, 0.0,-4.0},{-6.5, 2.5,-1.0},{-7.5, 2.0, 3.0},{-5.0, 2.5,-0.5},
{-2.5, 0.5,-3.0},{-2.0, 1.0,-1.5},{-3.0, 2.0, 7.0},{-0.5,-0.5,-2.0},
{-3.0,-3.0,-4.0},{-4.5,-3.5,-7.5},{-3.5,-2.5,-3.5},{-0.5, 0.5,-2.0},
{-3.5, 2.5,-3.5},{-4.5, 4.0,-7.5},{-3.0, 3.0,-4.0},{-2.0, 1.0,-4.0},
{ 0.0, 0.5,-2.5},{ 2.0, 1.0,-4.0},{ 0.5, 1.5, 7.5},{-0.5, 1.5, 7.5},
};

PT_PAIR _ln3[N_LN3]={
{ 1, 2},{ 2, 3},{ 3, 0},{ 4, 5},{ 5, 6},{ 6, 7},{ 7, 4},{ 8, 9},
{ 7,10},{10,11},{11,12},{12, 1},{ 4,12},{13,14},{14,15},{15,16},
{16,13},{17,18},{18,19},{19,17},{12,20},{20,21},{ 2,20},{20, 3},
{22,23},{23,24},{24,25},{25,22},{26,27},{27,28},{28,29},{29,26},
{ 6,10},{30,31},{31,32},{32,33},{33,30},{33,34},{34,35},{35,21},
{21, 0},{30,21},{36,37},{37,38},{38,39},{39,36},{40,41},{41,42},
{42,40},{43,44},{44,45},{45,46},{46,43},{47,48},{48,49},{49,50},
{50,47},{32,34},{35,51},{51,32},{51,52},{11,53},{53, 6},{53,52},
{52,20},{ 1,54},{54,55},{55, 0},
};

#define N_PT4   19
#define N_LN4   19

POINT_3D _obj4[N_PT4]={
{-0.6,-0.8, 1.2},{ 0.6,-0.8, 1.2},{ 0.0, 0.8,-1.8},{-1.0,-0.4,-1.6},
{-0.6, 1.2,-2.2},{-0.2, 1.6,-1.4},{-1.6, 0.8, 1.2},{-2.4,-1.8, 2.0},
{-1.0, 0.4,-0.2},{ 0.0,-0.2, 0.2},{ 1.0, 0.4,-0.2},{ 1.4,-0.2, 0.0},
{ 0.0,-0.4, 0.4},{-1.4,-0.2, 0.0},{ 1.0,-0.4,-1.6},{ 0.6, 1.2,-2.2},
{ 0.2, 1.6,-1.4},{ 1.6, 0.8, 1.2},{ 2.4,-1.8, 2.0},
};

PT_PAIR _ln4[N_LN4]={
{ 0, 1},{ 1, 2},{ 2, 0},{ 7, 3},{ 3, 4},{ 4, 5},{ 5, 6},{ 6, 7},
{ 8, 9},{ 9,10},{10,11},{11,12},{12,13},{13, 8},{18,14},{14,15},
{15,16},{16,17},{17,18},
};

#define N_PT5   27
#define N_LN5   31

POINT_3D _obj5[N_PT5]={
{-0.8,-0.8, 2.4},{-0.8,-0.8, 0.6},{-0.4, 0.0,-0.2},{-0.4, 0.0,-1.4},
{-1.6,-0.6, 2.4},{-2.0,-0.2, 0.4},{-1.2, 0.4,-2.2},{-0.6,-0.4, 1.6},
{-1.2, 0.4, 1.4},{ 0.8,-0.8, 2.4},{ 0.8,-0.8, 0.6},{ 0.4, 0.0,-0.2},
{ 0.4, 0.0,-1.4},{ 1.6,-0.6, 2.4},{ 2.0,-0.2, 0.4},{ 1.2, 0.4,-2.2},
{ 0.6,-0.4, 1.6},{ 1.2, 0.4, 1.4},{-0.4,-0.4, 2.0},{ 0.4,-0.4, 2.0},
{ 0.0, 0.4,-2.2},{-0.6, 0.4, 1.4},{-0.4, 0.2, 1.4},{-0.2, 0.6,-2.6},
{ 0.6, 0.4, 1.4},{ 0.4, 0.2, 1.4},{ 0.2, 0.6,-2.6},
};

PT_PAIR _ln5[N_LN5]={
{ 0, 1},{ 1, 2},{ 2, 3},{ 0, 4},{ 4, 5},{ 5, 6},{ 3, 6},{ 6, 1},
{ 0, 7},{ 7, 2},{ 4, 8},{ 8, 6},{ 9,10},{10,11},{11,12},{ 9,13},
{13,14},{14,15},{12,15},{15,10},{ 9,16},{16,11},{13,17},{17,15},
{18,19},{19,20},{20,18},{22,23},{23,21},{25,26},{26,24},
};

#define MAX_OBJ 5          /* 物体の最大数 */
#define MAX_EM  100        /* 要素(頂点・辺)の最大数 */
OBJECT  object[MAX_OBJ];
/*
        ワークエリア
*/
POINT_2D w_obj1[ MAX_OBJ * MAX_EM];      /* 描画用 */
POINT_2D w_obj2[ MAX_OBJ * MAX_EM];      /* 消去用 */
/*
        定数
*/
double  dist1=100.0;     /* Z軸上,視点から原点までの距離 */
double  dist2=2.0;       /* Z軸上,視点から投影面までの距離 */
int     doLINE;          /* 実際にラインを引くかを示す */
#ifndef PI
#define PI      3.14159265
#endif

#ifndef V70
/*
        平行移動
*/
Slide(obj,dx,dy,dz)
OBJECT  *obj;
double  dx;
double  dy;
double  dz;
{
        int     i,n;
        n=obj->np;
        for(i=0;i<n;i++){
                obj->pt[i].x += dx;
                obj->pt[i].y += dy;
                obj->pt[i].z += dz;
        }
}
/*
        X軸回りの回転
*/
RotateByX(obj,ang)
OBJECT  *obj;
double  ang;
{
        double  s,c;
        double  y,z;
        int     i,n;
        s=sin(ang);
        c=cos(ang);
        n=obj->np;
        for(i=0;i<n;i++){
                y =  (obj->pt[i].y)*c-(obj->pt[i].z)*s;
                z =  (obj->pt[i].y)*s+(obj->pt[i].z)*c;
                obj->pt[i].y = y;
                obj->pt[i].z = z;
        }
}
/*
        Y軸回りの回転
*/
RotateByY(obj,ang)
OBJECT  *obj;
double  ang;
{
        double  s,c;
        double  x,z;
        int     i,n;
        s=sin(ang);
        c=cos(ang);
        n=obj->np;
        for(i=0;i<n;i++){
                x =  (obj->pt[i].x)*c+(obj->pt[i].z)*s;
                z = -(obj->pt[i].x)*s+(obj->pt[i].z)*c;
                obj->pt[i].x = x;
                obj->pt[i].z = z;
        }
}
/*
        Z軸回りの回転
*/
RotateByZ(obj,ang)
OBJECT  *obj;
double  ang;
{
        double  s,c;
        double  x,y;
        int     i,n;
        s=sin(ang);
        c=cos(ang);
        n=obj->np;
        for(i=0;i<n;i++){
                x = (obj->pt[i].x)*c-(obj->pt[i].y)*s;
                y = (obj->pt[i].x)*s+(obj->pt[i].y)*c;
                obj->pt[i].x = x;
                obj->pt[i].y = y;
        }
}
/*
        3D→2D変換(適当にスケーリングして整数化)
*/

TransTo2D(obj)
OBJECT  *obj;
{
        double  scale;
        int     i,n;
        POINT_2D *tmp;
        n=obj->np;
        tmp = obj->wk2;
        obj->wk2 = obj->wk1;
        obj->wk1 = tmp;
        for(i=0;i<n;i++){
                scale = dist2/(dist1-(obj->pt[i].z));
                obj->wk1[i].x = scale*(obj->pt[i].x)*768.0+384.0;
                obj->wk1[i].y = scale*(obj->pt[i].y)*512.0+256.0;
        }
}
#endif

/*
        グラフィック表示
*/
InitScreen()
```

▶とうとう夏がきてしまいました。夏にはときどき，コンピュータが熱暴走をしてしまうので困ってしまいます。小さい扇風機は絶対に必要ですね。　　内間　正晃(20)静岡県

```
{
        CRTMOD(16);
        G_CLR_ON();
}
#ifndef V70
Display(obj)
OBJECT  *obj;
{
#ifdef  DEBUG
        struct LINEPTR LP0,LP1;
#endif
        struct LINEPTR *lp0,*lp1;
        int     i,n;
        int     fr,to;
        n=obj->nl;
#ifdef  DEBUG
        lp0= &LP0;
        lp1= &LP1;
#else
        lp0=(struct LINEPTR*)0xb1f400;
        lp1=(struct LINEPTR*)0xb1f40c;
#endif
        for(i=0;i<n;i++){ /* 表示 */
                fr = obj->ln[i].fr_point;
                to = obj->ln[i].to_point;
                lp0->x1 = obj->wk2[fr].x;
                lp0->y1 = obj->wk2[fr].y;
                lp0->x2 = obj->wk2[to].x;
                lp0->y2 = obj->wk2[to].y;
                lp0->color = 0;
                lp0->linestyle = -1;
                lp1->x1 = obj->wk1[fr].x;
                lp1->y1 = obj->wk1[fr].y;
                lp1->x2 = obj->wk1[to].x;
                lp1->y2 = obj->wk1[to].y;
                lp1->color = obj->color;
                lp1->linestyle = -1;
                if(doLINE){
#ifdef DEBUG
                        LINE( lp0 );
                        LINE( lp1 );
#else
                        ROMVER();
#endif
                }
        }
        KEYSNS();       /* ブレークチェック */
}
#endif
/*
        データの初期化
*/
InitData()
{
        object[0].pt  = _obj1;
        object[0].ln  = _ln1;
        object[0].wk1 = &w_obj1[MAX_EM*0];
        object[0].wk2 = &w_obj2[MAX_EM*0];
        object[0].np  = N_PT1;
        object[0].nl  = N_LN1;
        object[0].color= 15;

        object[1].pt  = _obj2;
        object[1].ln  = _ln2;
        object[1].wk1 = &w_obj1[MAX_EM*1];
        object[1].wk2 = &w_obj2[MAX_EM*1];
        object[1].np  = N_PT2;
        object[1].nl  = N_LN2;
        object[1].color= 9;

        object[2].pt  = _obj3;
        object[2].ln  = _ln3;
        object[2].wk1 = &w_obj1[MAX_EM*2];
        object[2].wk2 = &w_obj2[MAX_EM*2];
        object[2].np  = N_PT3;
        object[2].nl  = N_LN3;
        object[2].color= 13;

        object[3].pt  = _obj4;
        object[3].ln  = _ln4;
        object[3].wk1 = &w_obj1[MAX_EM*3];
        object[3].wk2 = &w_obj2[MAX_EM*3];
        object[3].np  = N_PT4;
        object[3].nl  = N_LN4;
        object[3].color= 7;

        object[4].pt  = _obj5;
        object[4].ln  = _ln5;
        object[4].wk1 = &w_obj1[MAX_EM*4];
        object[4].wk2 = &w_obj2[MAX_EM*4];
        object[4].np  = N_PT5;
        object[4].nl  = N_LN5;
        object[4].color= 5;

        Slide(&object[0],0.0,-7.0,14.0);            /* 位置合わせ */
        Slide(&object[1],15.0,12.0,-11.0);
        Slide(&object[2],-6.0,11.0,-8.0);
        Slide(&object[3],-15.0,6.0,7.0);
        Slide(&object[3],4.0,-2.0,-10.0);
}
/*
        それぞれの物体の動きを定義する関数
*/
static  double  px0=0,py0=0,pz0=0;
static  double  px1=0,py1=0,pz1=0;
static  double  rad=0;

action0(obj)
OBJECT  *obj;
{
        Slide(obj,px1-px0,py1-py0,pz1-pz0);
        if(rad<(0.5*PI))
                RotateByZ(obj,PI/90.0);
        else if(rad<(1.0*PI))
                RotateByY(obj,PI/180.0);
        else
                RotateByX(obj,PI/90.0);
        rad += PI/180.0;
        if(rad>=(1.5*PI)) rad=0;
        px0=px1;
        py0=py1;
        pz0=pz1;
        px1=1.2*cos(rad)-1.2;
        py1=1.2*sin(rad);
        pz1=2.0*sin(rad);
        TransTo2D(obj);
        Display(obj);
}

action1(obj,ang)
OBJECT  *obj;
double  ang;
{
        RotateByX(obj,ang);
        TransTo2D(obj);
        Display(obj);
}

action2(obj)
OBJECT  *obj;
{
        RotateByX(obj,PI/90.0);
        RotateByY(obj,PI/90.0);
        RotateByZ(obj,PI/90.0);
        TransTo2D(obj);
        Display(obj);
}

action3(obj)
OBJECT  *obj;
{
        RotateByZ(obj,PI/180.0);
        RotateByY(obj,PI/180.0);
        RotateByX(obj,PI/180.0);
        TransTo2D(obj);
        Display(obj);
}

/*
        メインプログラム
*/
main(argc,argv)
int     argc;
char    *argv[];
{
        int     i;
        int     start,end;
        double  px0,py0,pz0,px1,py1,pz1;

        B_SUPER(0);
        doLINE=(argc==1)? 1 : 0;
        InitData();
        InitScreen();
        start=ONTIME();
        for(i=0;i<600;i++){
                action0(&object[0]);
                action1(&object[1],PI/90.0);
                action1(&object[2],PI/180.0);
                action2(&object[3]);
                action3(&object[4]);
        }
        end=ONTIME();
        printf("Total %.1f sec.¥n",((double)(end-start))/100.0);
}
```

リスト2 DISPLAY.S

```
S_POINT_3D      .set    24
S_POINT_2D      .set    8
S_PT_PAIR       .set    8
S_LINEPTR       .set    12
#
#       1回で2度ライン描画(Display)をする版
#       かつ共有RAM領域を直接操作する版
#
        .globl  _Slide
        .text
        .align  4
_Slide:
        mov.w   [ap],r1         -- *obj
        fmov3v.l 4[ap],fr0
        mov.d   [r1],r0         -- r0=n,r1= *pt
SLOOP:
        fmov3v.l [r1],fr4
        fadd3v.l fr0,fr4
        fmov3v.l fr4,[r1]
        add.w   #S_POINT_3D,r1
        dbr     r0,SLOOP
        ret     #0

        .globl  _RotateByX
        .text
        .align  4
_RotateByX:
        mov.w   [ap],r1         -- *obj
        fmov.l  4[ap],fr25
        fsincos.l fr25,fr26     -- fr25;sin,fr26:cos
        mov.d   [r1],r0         -- r0=np,r1= *pt
#       ┌                       ┐
#       |1    0         0       |
#       |0    cos(x)   -sin(x)  |
#       |0    sin(x)   cos(x)   |
#       └                       ┘
        fcvt.wl #1,fr16         -- 1
        fcvt.wl #0,fr17         -- 0
        fmov.l  fr17,fr18       -- 0
        fmov.l  fr17,fr20       -- 0
        fmov.l  fr26,fr21       -- cos(x)
        fneg.l  fr25,fr22       -- -sin(x)
        fmov.l  fr17,fr24       -- 0
#       fmov.l  fr25,fr25       -- sin(x)
#       fmov.l  fr26,fr26       -- cos(x)
#
```

▶なぜ，ホクロに生えている毛は長いのだろう。　小山　優一(18)東京都

```
RXLOOP:
        fmov3v.l [r1],fr0
        fmul3v.l fr0,fr0
        fmov3v.l fr0,[r1]
        add.w   #S_POINT_3D,r1
        dbr     r0,RXLOOP
        ret     #0

        .globl  _RotateByY
        .text
_RotateByY:
        mov.w   [ap],r1         -- *obj
        fmov.l  4[ap],fr18
        fsincos.l fr18,fr16     -- fr18;sin,fr16:cos
        mov.d   [r1],r0         -- r0=np,r1=*pt
#               ┌                         ┐
#               │ cos(x)    0    sin(x)   │
#               │ 0         1    0        │
#               │-sin(x)    0    cos(x)   │
#               └                         ┘
#       fmov.l  fr16,fr16       -- cos(x)
        fcvt.wl #0,fr17         -- 0
#       fmov.l  fr18,fr18       -- sin(x)
        fmov.l  fr17,fr20       -- 0
        fcvt.wl #1,fr21         -- 1
        fmov.l  fr17,fr22       -- 0
        fneg.l  fr18,fr24       -- -sin(x)
        fmov.l  fr17,fr25       -- 0
        fmov.l  fr16,fr26       -- cos(x)
RYLOOP:
        fmov3v.l [r1],fr0
        fmul3v.l fr0,fr0
        fmov3v.l fr0,[r1]
        add.w   #S_POINT_3D,r1
        dbr     r0,RYLOOP
        ret     #0

        .globl  _RotateByZ
        .text
        .align  4
_RotateByZ:
        mov.w   [ap],r1         -- *obj
        fmov.l  4[ap],fr20
        fsincos.l fr20,fr21     -- fr20;sin,fr21:cos
        mov.d   [r1],r0         -- r0=np,r1=*pt
#               ┌                        ┐
#               │cos(x)   -sin(x)    0   │
#               │sin(x)    cos(x)    0   │
#               │0         0         1   │
#               └                        ┘
        fmov.l  fr21,fr16       -- cos(x)
        fneg.l  fr20,fr17       -- -sin(x)
        fcvt.wl #0,fr18         -- 0
#       fmov.l  fr20,fr20       -- sin(x)
#       fmov.l  fr21,fr21       -- cos(x)
        fmov.l  fr18,fr22       -- 0
        fmov.l  fr18,fr24       -- 0
        fmov.l  fr18,fr25       -- 0
        fcvt.wl #1,fr26
RZLOOP:
        fmov3v.l [r1],fr0
        fmul3v.l fr0,fr0
        fmov3v.l fr0,[r1]
        add.w   #S_POINT_3D,r1
        dbr     r0,RZLOOP
        ret     #0

        .globl  _TransTo2D
        .text
        .align  4
_TransTo2D:
        mov.w   [ap],r0         -- *obj
        mov.d   16[r0],r2       -- r2=obj->wk1[0],r3=obj->wk2[0]
        mov.w   r2,20[r0]
        mov.w   r3,16[r0]       --<r3> new obj->wk1[0]
        mov.d   [r0],r0         --<r0> np, <r1> obj->pt[0]
        fcvt.wl #768,fr16       -- const
        fcvt.wl #384,fr17
        fcvt.wl #512,fr18
        fcvt.wl #256,fr19
        fmov.l  _dist2,fr20
        fmov.l  _dist1,fr21
T2LOOP:
        fmov.l  fr20,fr0
        fmov.l  fr21,fr1
        fsub.l  16[r1],fr1
        fdiv.l  fr1,fr0         -- scale
        fmov.l  fr0,fr1
        fmul.l  [r1],fr0
        fmul.l  fr16,fr0
        fadd.l  fr17,fr0
        fmul.l  8[r1],fr1
        fmul.l  fr18,fr1
        fadd.l  fr19,fr1
        fcvt.lw fr0,[r3]
        fcvt.lw fr1,4[r3]
        add.w   #S_POINT_3D,r1
        add.w   #S_PT_PAIR,r3
        dbr     r0,T2LOOP
        ret     #0

        .globl  _Display
        .text
        .align  4
_Display:
        prepare #(S_LINEPTR*2)
        pushm   #0m<r9-r21>
        mov.w   [ap],r0         -- *obj
        mov.w   8[r0],r21       -- nl
        mov.d   12[r0],r11      -- r11=ln[],r12=wk1
        mov.d   20[r0],r13      -- r13=wk2,r14=color
        movs.hw #0xf400,r15     -- lp0
        movs.hw #0xf40c,r16     -- lp1
        rot.h   #8,r14          -- バイト順序を変更
        mov.h   #0,8[r15]       -- カラーコード
        not.h   #0,10[r15]      -- ラインスタイル
        mov.h   r14,8[r16]      -- カラーコード
        not.h   #0,10[r16]      -- ラインスタイル
DLOOP:
        mov.d   [r11],r17       -- r17=fr,r18=to
        mov.d   [r13](r17),r19  -- wk2[fr].x,wk2[fr].y
        rot.h   #8,r19
        mov.h   r19,[r15]       -- x1
        rot.h   #8,r20
        mov.h   r20,2[r15]      -- y1
        mov.d   [r13](r18),r19  -- wk2[to].x,wk2[to].y
        rot.h   #8,r19
        mov.h   r19,4[r15]      -- x2
        rot.h   #8,r20
        mov.h   r20,6[r15]      -- y2
#
        mov.d   [r12](r17),r19  -- wk2[fr].x,wk2[fr].y
        rot.h   #8,r19
        mov.h   r19,[r16]       -- x1
        rot.h   #8,r20
        mov.h   r20,2[r16]      -- y1
        mov.d   [r12](r18),r19  -- wk2[to].x,wk2[to].y
        rot.h   #8,r19
        mov.h   r19,4[r16]      -- x2
        rot.h   #8,r20
        mov.h   r20,6[r16]      -- y2
#
        test.w  _doLINE
        je      DSKIP
#
#       ROMバージョン処理をフックしてある
#
        movz.bw #0x8f,r0        -- ROMVER
        trap    #0xa1
DSKIP:
        add.w   #S_PT_PAIR,r11
        dbr     r21,DLOOP
        call    _KEYSNS,[sp]
        popm    #0m<r9-r21>
        dispose
        ret     #0
#
#       B_SUBERもどき
#
#       ユーザーモードには戻れない！
#
        .globl  _B_SUPER
        .text
        .align  4
_B_SUPER:
        pushm   #0m<r0-r30>
        mov.w   sp,r30
        movs.hw #0xe060,r9
        movz.bw #0x84,r0        -- B_LPEEK
        trap    #0xa1
        mov.w   r0,r29
        movea.w h1,r1
        movs.hw #0xe060,r9
        movz.bw #0x88,r0        -- B_LPOKE
        trap    #0xa1
        chlvl   #0,#0           -- レベル0へ
        .align  4
h1:
        getpsw  r0
        and.w   #0x7fffffff,r0  -- 最上位ビットをクリア
        push    r0
        movea.w h2,[-sp]
        retis   #0
h2:
        mov.w   r29,/0xffffe060 -- ベクタを元に
        mov.w   r30,sp
        popm    #0m<r0-r30>
        ret     #0
```

リスト3 LINE.S

```
**********************************************************************
*        線分描画ルーチン
*
* このプログラムはＩＯＣＳのＲＯＭバージョン処理をフックして
* 別のライン処理（２つのラインを描く）を行う。
* 引数はＶ７０ボードとの共有領域に置いてあるものとする
*
**********************************************************************

        .text
        .even

BEGIN:
*
*         新しいＲＯＭＶＥＲ処理（ＬＩＮＥを２度呼ぶ）
*
gline:
        move.l  a1,-(sp)
        movea.l #$b1f400,a1  * 引数領域（その１）
line1:
        jsr     $00000000    *←このアドレスを書き換える
        movea.l #$b1f40c,a1  * 引数領域（その２）
line2:
        jsr     $00000000    *←このアドレスを書き換える
        move.l  (sp)+,a1
        rts

END:
START:
        move.w  #$1b8,-(sp)
        .dc.w   $ff35        * INTVCG（LINE処理アドレス獲得）
        addq.l  #2,sp

        move.l  d0,line1+2
        move.l  d0,line2+2

        pea     BEGIN(pc)
        move.w  #$18f,-(sp)
```

```
        .dc.w   $ff25           * INTVCS (ROMVER処理を変更)
        addq.l  #6,sp

        clr.w   -(sp)
        move.l  #(END-BEGIN),-(sp)
        dc.w    $ff31           * KEEPPR (常駐終了)

        .end    START
```

リスト4

```
10 fp=fopen("v70ipl.v70","rw")
20 fseek(fp,&h48a5,0)
30 fputc(&hc0,fp)
40 fputc(&h80,fp)
50 fputc(&h60,fp)
60 fputc(&h06,fp)
70 fputc(&hcd,fp)
80 fclose(fp)
```

リスト5

```
  10 str s[256],t[256],cr
  11 cr=chr$(13)+chr$(10)
  20 int n=1
  30 fpi=fopen("vfloat.s","r")
  31 fpo=fopen("vfloatp.s","c")
  40 while freads(s,fpi)<> -1
  49 t=s+cr
  50 fwrites(t,fpo)
  68 if s="L001fde:" then patch(): continue
  69 if s="L001ff4:" then patch(): continue
  70 if s="L00200a:" then patch(): continue
  80 if s="L002020:" then patch(): continue
  90 if s="L00203e:" then patch(): continue
 100 if s="L002054:" then patch(): continue
 110 if s="L00206c:" then patch(): continue
 120 if s="L002084:" then patch(): continue
 130 if s="L00209c:" then patch(): continue
 140 if s="L0020b4:" then patch(): continue
 150 endwhile
 160 fcloseall()
 170 end
1000 func patch()
1010 t=chr$(9)+"btst.b"+chr$(9)+"#5,16(sp)"+cr
1011 fwrites(t,fpo)
1020 t=chr$(9)+"bne"+chr$(9)+string$(n,"S")+cr
1021 fwrites(t,fpo)
1030 t=chr$(9)+"move.l"+chr$(9)+"usp,a6"+cr
1031 fwrites(t,fpo)
1032 t=string$(n,"S")+":"+cr
1040 fwrites(t,fpo)
1050 n=n+1
1060 endfunc
```

リスト6

```
        .globl  _pow
        .text
        .align  4
_pow:
        fmovcr  #0,fstw
        fmov.l  [ap],fr0
        fpower.l 8[ap],fr0
        fmovcr  fstw,[-sp]
        pop     r0
        tb      r0,no_err
        test1   #12,r0
        mov.w   #1,r1
        jne     err_fiv
        test1   #9,r0
        mov.w   #4,r1
        jne     err_fud
        test1   #10,r0
        mov.w   #3,r1
        jne     err_fov
no_err:
        ret     #0
        .align  4
err_fiv:
err_fud:
err_fov:
        fmov.l  fr0,[-sp]
        mov.d   8[ap],[-sp]
        mov.d   [ap],[-sp]
        push    #0x330
        push    r1
        call    _except,[sp]
        movea.w 0x20[sp],sp
        ret     #0
```

V70アクセラレータのデバッグ

前回，今回とV70ボードに関する記事を書いてきたわけですが，実際にボードを手にした人で動かないと悩んでいる人がいるかもしれません。実はV70ボードのソフトウェアパッケージにはいくつかのバグがあります。私はそれをデバッグしながらプログラムを書いていきました。ここで私が発見したバグに関する状況と対処法を報告しておきましょう。なお，私が使用したのはVDTK-68K(ボードパッケージ)，VDTK-C-68K（Cコンパイラ）ともver.1.01のものですから，今回のパッチはver.1.01のみに有効です。これ以降のバージョンの人は同様の症状がないか確認してください。もし誤動作したら発売元に問い合わせてください。

バグその1

●症状

VFLOAT.Xで__POWER（FE3F$_H$）のシステムコールがむちゃくちゃな値を返す。

●理由

V70IPL.V70の__POWER処理ルーチンが引数として不定の値を受け取っているため。

●デバッグ

V70IPL.V70を直接書き換える。カレントディレクトリにV70IPL.V70を置いて，X-BASICからリスト4のプログラムを実行する。

バグその2

●症状

VFLOAT.Xでスタックを介して引数を受け渡すシステムコールである，

__CDADD（FEED$_H$）
__CDSUB（FEEE$_H$）
__CDMUL（FEEF$_H$）
__CDDIV（FEF0$_H$）
__CDMOD（FEF1$_H$）
__CFADD（FEF3$_H$）
__CFSUB（FEF4$_H$）
__CFMUL（FEF5$_H$）
__CFDIV（FEF6$_H$）
__CFMOD（FEF7$_H$）

の値がむちゃくちゃである。このバグのためGCCなどが誤動作する。

●理由

VFLOAT.Xでの処理が引数がスーパーバイザスタックにあるものとして行われているため。それぞれの処理の先頭で，

```
    btst.b #5,16(sp)
    bne    SKIP
    move.l usp,a6
SKIP:
```

という処理が抜けている。

●デバッグ

変更箇所が多いので直接VFLOAT.Xを書き換えるということはできない。DIS.X（創刊8周年PRO-68Kや電脳倶楽部に掲載）でソースコードを生成して，それにパッチを当てる。

DIS VFLOAT.X VFLOAT.S

によって作られるVFLOAT.Sというソースファイルをカレントディレクトリに置き，X-BASICからリスト5のプログラムを実行すると，パッチが当てられたVFLOATP.Sというファイルが生成される（エディタで直接修正したほうが早いという説もある）。これをアセンブル，リンクするとVFLOATP.Xというファイルができるので，VFLOAT.Xにリネームして使用する。

バグその3

●症状

AFPP用の数学ライブラリ（COPRO.A）を使用する場合，pow関数の第1引数と第2引数が逆になっている。当然，結果を誤る。

●原因

単なるケアレスミスのような気がする。真相はわからない。

●デバッグ

ライブラリを変更する。リスト6のプログラムをpow.sというファイル名で作成し，V70ボード付属のVAS.Xでアセンブルしてpow.oというファイルを作る。これを，同じくV70ボードに付属のVLIB.Xによりライブラリを変更する。カレントディレクトリにCOPRO.Aとpow.oを置き，

VLIB /R COPRO.A pow.o

を実行するとライブラリが変更される。

バグその4

●症状

Cコンパイラ（VCC）でスタティックなローカル変数を使用するとき，最適化オプションを付けると，変数の領域がソースプログラムから消されてしまう。このためリンク時にエラーになる。

●理由，デバッグ

おそらくは，変数領域が分岐命令で参照しないラベル（不要なラベル）として削除されてしまうのだろう。修正はコンパイラを作ったハドソン以外にはできない。

バグその5

●症状

Cコンパイラでプログラムが正しくコンパイルできない。特に構造体を使用した式のコード生成を誤ることが多い。たとえば，今回のリスト1のDisplay関数などのコード生成を誤る。

●理由，デバッグ

ハドソンさん，なんとかしてください。せめてXCのver.2.1並の品質にしてくれないと，C言語によるプログラム開発は不可能です。

68881の性能を引き出す

FPP.MACの作成

Taki Yasushi 瀧　康史

FLOAT3.Xを使うことで，実数演算を高速化してくれる数値演算プロセッサボード。ここでは，FLOAT3.Xによるオーバーヘッドをなくし，68881の性能をより引き出すためのマクロを作成します。

あなたは今日，秋葉原にいって数値演算プロセッサボード，もしくは68881を買ってきました。これで家のX68000もちょっとは速くなるかなと，喜びの気持ちで家に帰りX68000に接続してみました。最初から実数演算だけ速くなると割り切って買った人はともかく，こんな気持ちで買ってきた人は結構いると思います。でも，買ってみてその68881，いったいどうやって使うんでしょう。

もっとも一般的で，比較的X68000のシステム事情をよくわかっている人なら，まずFLOAT2.XをFLOAT3.Xに変えてみて，いろいろなソフトを実行してみると思います。ゲームの中に入っているFLOAT2.Xを片っ端からFLOAT3.Xに変えた人もいることでしょう。結果はどうですか？　いろんなソフトが速くなりました？　私の身の周りにあるソフトで，劇的に変化したものは「CANVAS PRO-68K」。それから……というように，探しても目に見えて速くなるものってあんまりないんですよ。

Cで作られているソースは，一応すべてFラインファンクションコール（FLOATのコール）を使っているのですが，数値演算プロセッサが使われてかなりのスピードアップが望めるものは実数演算だけ。だから「CANVAS PRO-68K」のような実数演算ばっかりのソフトは，比較的スピードアップが望めますが，現在あなたが使っているワープロやゲーム，通信ソフトなどは，人間の目で見る限り，まず速くなっていないと感じるでしょう。

また，よく86系はコプロを差すだけでスピードが速くなる，と勘違いしてる人がいますが，これはソースがあって，コプロありのオプションを付けて再コンパイルした場合に限ります。UNIXのようにソフトウェアをソースレベルで配布して，個々のマシンでコンパイルするものならともかく*1，現時点でのパソコンのように，そのほとんどが実行形式のみで配布されている場合（特に市販品に多い）には，効果がないといってよいでしょう。

それに比べて，Human68kでは数値演算はファンクションコールによって行いますので，市販ソフトのようなすでに実行形式のみで配布されているソフトも，数値演算プロセッサを使ったファンクションコールなどに代えたり，ファンクションルーチンの改善によって実数演算部分のみの高速化が可能になります。

しかし，この方法では，数値演算に対してある一定のファンクションコールを呼び出すため，その呼び出し部分でのオーバーヘッドが数値演算のたびにかかります。また，数値演算プロセッサを使うときに，その数値演算プロセッサ内のレジスタを効果的に使った演算ができないため，数値演算プロセッサの真の機能が発揮できないのです。

以上のような理由で，FLOAT3.Xに入れ替えても，期待に応えるようなスピードアップが望めず，結局お金をかけたわりには意味がなかったとか，ひどい場合には数値演算プロセッサボードを持っているにもかかわらず，スロットの関係で展示品になってしまう場合もありうるのです。

しかし，数値演算プロセッサ自身は，かなり性能がよいものです。実際，FLOAT3.Xを通さず数値演算プロセッサを直接アクセスした場合は，かなりのスピードアップが望めます。それを実証するためにも，とりあえず，7月号のDōGA CGAシステムの中にあるrendxvi.xをrend.xにリネームしてみてください。そこでお試しシステムを立ち上げ，ワゴンの編隊飛行でも，なんでもいいんですから，アニメーションを作ってみましょう。もちろん，FLOAT2.XはFLOAT3.Xに代えておくことも忘れないでください。

どうですか？　1回やったものと同じものをレンダリングするのがお勧めなのですが，目に見えてスピードアップがわかるでしょう。数値演算プロセッサもちゃんと使ってやれば，それなりに動作することがわかってもらえましたか？

---

*1　マシンの利用者がすべて技術者ならともかく，パソコンのようなユーザーフレンドリであるべきものには，この用法は必ずしもよいとはいえません。

68881の性能

68881そのものの性能はいろいろな書籍によって参照できるので，ここではざっと紹介程度ですませます。

68881には，全部でFP0～FP7まで80ビット長のレジスタが8本，コントロールレジスタとして，32ビット長のFPCR,FPSR,FPAIR（このFPAIRは外部バスとして接続しているX68000には意味がない）の3本があります。コプロセッサとして接続する場合には，あたかもCPUにレジスタがこの分増えたように使えるのですが，X68000では外部バスを通すため*2，それなりに使い方が限定されます。

68000で扱えるレジスタのデータフォーマットは，基本的にByte,Word,Longwordの3つですが，68881ではこのほかに，

1)　Short Real
　23ビット小数部，8ビット指数部，1ビット小数符号
2)　Double Real
　52ビット小数部，11ビット指数部，1ビット小数符号
3)　Expand Real
　64ビット仮数，1ビット小数点，15ビット指数部，1ビット仮数部符号
4)　Packed Decimal Real
　17ケタ仮数部，11ビット指数部，2ビット無限大フラグ，1ビット指数符号，1ビット仮数符号

と，合計7つ（図1）のデータフォーマッ

トがあるのです。なお，Short Real, Double Real, Expand Realのデータフォーマットはそれぞれ IEEE規格に準じてます。

データフォーマットを見てわかるように扱える数字には，計算して得られるそれぞれの桁数の数字限界のほかに，無限大（最高指数の限界値を超えたときのオーバーフロー），そして非数（無限大÷無限大）などが扱えます。

これらのデータフォーマットは，68881外部とアクセスするときに利用されるものであって，68881内部ではすべての演算が拡張精度で行われています（図2）。よってレジスタ↔レジスタ間演算（転送も含む）は，拡張精度しか存在しません。この場合でいうレジスタとは，68881のレジスタであって，68000のものではありません。そして，外部とはメモリや68000のレジスタなどを指します。

＊2　68000には，コプロセッサをハード的に接続できません。バスを通してI/Oデバイスとして利用します。

68881を叩く

問題なのはこの数値演算プロセッサを直接アクセスする（俗語：叩く）には，結構面倒くさいことをしなくてはならないのです。メーカーからも，数値演算プロセッサを使う方法については提示されてませんし（FLAOT3.X を使ってください，ということですかね），資料のほうも，最近やっと「Inside X68000」が解説し直してくれたくらいです。あとは，1988年8月号の本誌特集で68881を叩くそれなりのマクロを掲載した，その程度でしょうか。

数値演算プロセッサの命令は68020で使った場合，数値演算プロセッサ命令を直接アセンブラのソースと交ぜて（あたかも68020に命令が増えたかのように）使えるのですが，68000ではバス接続のため，そうもいきません。

X68000が命令を68881に与えるためには，CIRと呼ばれる68881との接続ポートをある一定の順序でアクセスします。これは使用する命令（というかオペランド）によって違います。たとえば，68881のレジスタ→68020のレジスタ間の命令には，（68881のアセンブラで表記すると）次のようなものがあります。ちなみに，命令の最初にFが付くのはコプロ命令です（68020に接続時の名称）。

```
FMOVE.X FP0,D0
```

このFMOVEはデータを加工せずにFP0→D0へ転送する命令です。この「FMOVE.X FP0,D0」をX68000で実行するには，次の手順を踏まなくてはなりません。

1)　コマンドCIRにコマンドコード＊3を書き出す。

2)　68881からの応答コード（ヌルプリミティブコード）を得るまでウェイトを入れる（この時間は微小）。

3)　この間MC68881は計算するので，しばし，応答コード（データ転送プリミティブ）がくるまで待つ（これも微小）。

4)　オペランドCIRから，D0レジスタに入れるべき内容を読み出す。

以上のようにデータを取り出す命令でも，このような4ステップの手順を踏まなくてはなりません。そのうえ，68881に送るコマンドコードを求めるのが，なかなか大変です（当然ですが，デスティネーションのレジスタ，データフォーマットなどをコマンドコードに組み込む必要がある）。そのため，ある一定の手順を連ねたマクロを作り，それに併せてそのマクロを実行すれば効果的です。ということで，数値演算プロセッサを直接叩くためのマクロを考えてみましょう。

＊3　コマンドコードとは，それぞれのコプロ命令の命令形式の2ワード目です（コプロ命令はすべて2ワード以上の命令）。1ワード目はコプロID，実行アドレス表記のため，X68000のような外部バスを通したものの場合，無意味になります。

図1　68881が扱うことができるデータのフォーマット

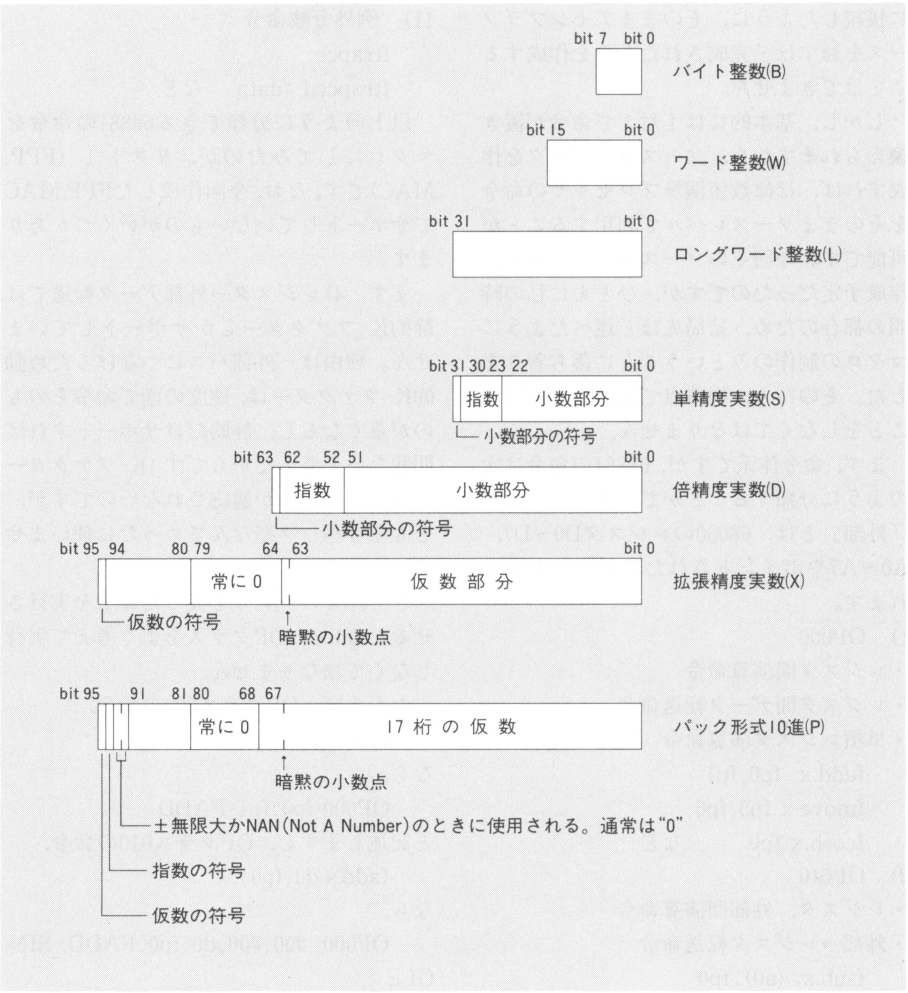
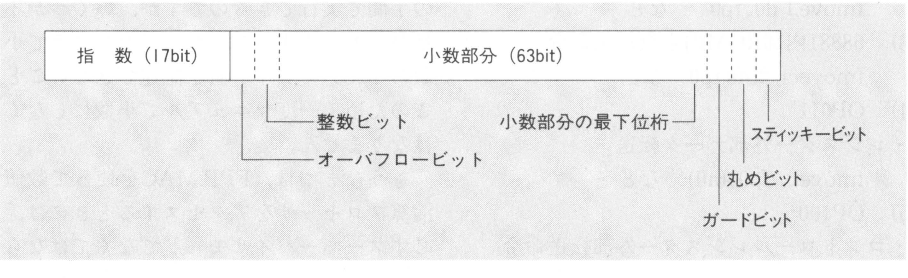
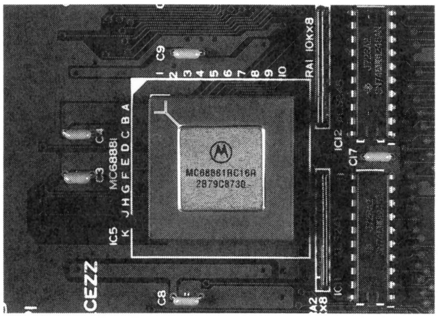


図2　68881内部での演算途中のフォーマット





これが，今回ターゲットにした68881

68881のマクロ

アセンブラのマクロといっても，それほど強力なわけではないので，あたかも68020に接続したように，そのままアセンブラソースを通すほど完成されたものを作成することはできません。

しかし，基本的には1対1で命令が置き換えられますから，ソースコンバータを作成すれば，ほぼ数値演算プロセッサの命令をそのままソースレベルで利用することが可能です。今回そのソースコンバータも，作成予定だったのですが，ひとえに私の時間の都合のため，結局先ほど述べたようにマクロの制作のみということに落ち着きました。そのため，現時点ではかなり面倒なことをしなくてはなりません。

まず，命令体系ですが，68881の命令は次のように分類することができます。ここで「外部」とは，68000のレジスタD0〜D7，A0〜A7やポイントされたアドレスも含まれます。

1) OP000
・レジスタ間演算命令
・レジスタ間データ転送命令
・単項レジスタ演算命令
　fadd.x fp0,fp1
　fmove.x fp5,fp6
　fcosh.x fp0　　など
2) OP010
・レジスタ，外部間演算命令
・外部→レジスタ転送命令
　fsub.x (a0),fp0
　fsinh.l d0,fp0
　fmove.l d0,fp0　など
3) 68881内部ROM内定数転送命令
　fmovecr #ofs,fp0　など
4) OP011
・レジスタ→外部データ転送
　fmove.x fp0,(a0)　など
5) OP100
・コントロールレジスタ→外部転送命令
　fmove.l fpcr,d0　など
6) OP101
・外部→コントロールレジスタ転送命令
　fmove.l d0,fpsr　など
7) 複数浮動小数点データレジスタ転送命令
　fmovem.x fp0-fp7,-(SP)
　fmovem.x (SP)+,fp0-fp7　など
8) OP011,OP100の複数レジスタ版
　fmovem.x fpcr-fpsr,+(SP)　など
9) 条件付き分岐命令
　fbcc　label
　fdbcc　Dn,label　など
10) FSAVE/FRESTORE命令
　fsave　-(a0)
　frestore (a0)+
11) 例外分岐命令
　ftrapcc
　ftrapcc.l #data　など

以上のように分類できる68881の命令をマクロにしてみたのが，リスト1（FPP.MAC)です。なお，今回作成したFPP.MACでサポートしていないものがいくつかあります。

まず，4)レジスタ→外部データ転送では，静的K-ファクターしかサポートしていません。理由は，外部バスにつなげるため動的K-ファクターは，速度の面で命令そのものが重くなるし，静的だけサポートすれば問題ないと考えたからです（K-ファクターはパック型にしか適応されないのですが，そもそもパック型なんてめったに使いません)。

さて，このマクロを使った命令を実行させるためには，OPクラスをよく考えて実行しなくてはなりません。

たとえば，OPクラス000の命令，
　fadd.x fp0,fp1
なら，
　OP000 fp0,fp1,FADD
と記述しますし，OPクラス010の命令，
　fadd.s d0,fp0
なら，
　OP000 #00,#00,d0,fp0,FADD,_SINGLE
と実行します。ほとんどの命令がこの程度の手間で実行できるのですが，いくつか不都合があります。ひとつはアセンブラで小数のイミディエイト値を記述できないこと。このため，一度マニュアルで小数にしなくはなりません。

もうひとつは，FPP.MACを使って数値演算プロセッサをアクセスするときには，必ずスーパーバイザモードでなくてはならないこと，です。これは，現時点のような，マクロによる命令の再現では，絶対に回避不可能なのでがまんしてください。

最後にこのFPP.MACで再現してない命令について。まず，4)レジスタ→外部データ転送で扱うことのできるFMOVEの動的K-ファクター。そして，7)，8)複数レジスタ転送命令，10)frestore，fsave，11)ftrapなどは，再現の方法が私には考えつきませんでした。FSccはマニュアルのいっていることがよくわからず，fnopは意味がないと思ったので再現していません。

サンプルプログラム

せっかく数値演算プロセッサを直接アクセスするマクロを作成したので，ちょっとしたサンプルプログラムを2つほど作成してみました。

まず，リスト2の__fsin.sというプログラムです。これは，FPP.MACを使って作った関数をCのプログラムに応用する簡単な例です。この例ではSIN関数だけの演算ですが，ソース中のFSINをFCOSに変えたりするだけで，COS関数のルーチンになります。

なお，引数はfloat，返り値もfloatですので，このルーチンを使うときには必ず，
　float　fsin(float);
という行を入れて使用してください。ソース中のsin関数を書き換えるだけで，思ったよりスピードアップしました。

もうひとつはリスト3で，比較的計算量の多いプログラムを作ってみました。こちらは1991年12月号の本誌に掲載された，石上氏のFFTによる音のスペクトルを得るプログラムのFFT( )関数の部分を，数値演算プロセッサを直接叩くアセンブラに書き直して見ました（リスト4に再掲載)。

すでに入力されている人は，あらかじめ，fft.sをアセンブルし，石上さんのプログラムからFFT( )関数を削除してコンパイルし，fft.oをリンクしてください。

実行結果ですが，もともと7〜10秒ぐらいの計算スピードが2秒弱にまで短縮できました。もっとも，プログラムをアセンブラに書き直したり，アルゴリズムを多少変えたことも大きいと思いますが，満足できる結果でしょう。

それでは，FPP.MACマクロマニュアルとこれらのサンプルプログラムを参照しながら使い方を把握していってください。本格的に知りたい方は，参考文献1)の「Inside X68000」を参照するといいでしょう。

68881とSX-WINDOW

さて，直接数値演算プロセッサを叩くことで,68881の性能をかなり引き出せることがおわかりいただけたと思います。

しかし，直接数値演算プロセッサを叩いたプログラムには，意外なところで盲点があります。それはシステムが数値演算プロセッサのことを考えてないことに起因する弊害なのですが,マルチタスクシステム(疑似でも）では，同時に複数の数値演算プロセッサを直接，もしくは間接的にアクセスするプログラムが実行できないのです。

通常マルチタスクシステムでは，タスクの切り替え時にレジスタをスタックにプッシュする必要があります。ところがSX-WINDOWや，OS-9，Human68k上でもスレッドを使ったバックグラウンドプログラムは，数値演算プロセッサのレジスタのことを考えて作られていないのです。まあ，当たり前[*4]といえば当たり前ですが，SX-WINDOWなどは今後発展する領域ですし，なんらかの処置をしてほしいところです。

OS-9やHuman68kのスレッドは，基本的にアプリケーション側でタスクチェンジがいつ行われるか,知る手段がありません[*5]。イベントドリブン型のSX-WINDOWでは，タスクチェンジが行われそうなAラインコール（SXコール)が予測できるので，それらの前にレジスタのセーブ，あとにリストアを入れればできなくもありません。しかし，非常に面倒ですしCから利用するのが難しいので，今後の検討事項でしょう。

＊4　先に述べたとおり，68881には実にたくさんのレジスタがあって，タスクチェンジ時にいちいちそれをチェックして転送作業を行うと，しこたま時間がかかってしまうからです。
＊5　タイマを使って強制的にタスクチェンジをするからです。

まとめ

せっかく買った数値演算プロセッサを活用させるのはあなた次第です。スロットの関係で，結局，数値演算プロセッサボードが飾り物に成り下がってしまうか，あってもFLOAT2.Xの少し速いものとしてしか使われないとか……それもひとつの使い方で私の立場では，ひとそれぞれ，としかいえませんが，それでもあなたの数値演算プロセッサが，少しでもあなたの役に立つように，と思って作成したFPP.MACが活用されると非常に嬉しいですね。

〈参考文献〉

1)　桒野雅彦：「Inside X68000」，ソフトバンク，1992
2)「MC68881ユーザーズマニュアル」，電波新聞社，1988
3)　石上達也：「冬の夜長のスペクトル解析」，Oh!X，1991年12月号

FPP.MACマクロマニュアル

●シンボル（A5reg)

シンボル(A5reg)が定義されているとCIRへのアクセスが，A5相対アドレッシングモードでアクセスされる。定義されていないと物理アドレスでアクセスする。これは，68000がI/Oアクセス時に物理アドレスでアクセスするより，アドレスレジスタ相対アドレッシングでアクセスしたほうが速いために用意したもの。定義すると1命令のたびに4クロック～12クロックほど高速化されるが，A5レジスタはプログラム全体を通して使用できなくなる。

●マクロ中で定義されるシンボル

マクロ中で主に使用されて，通常は使われないもの。

CIRADR	CIRがあるベースアドレス
CIRRES	応答CIR
CIRCON	コントロールCIR
CIRSAVE	セーブCIR
CIRREST	リストアCIR
CIRCMD	コマンドCIR
CIRCND	コンディションCIR
CIROPR	オペランド（データ）CIR
CIRRS	レジスタ選択CIR

●応答CIRの返り値（ヌルプリミティブ）

circc	コンディションCIRへの書き込みに対する応答（真）
cirnull	68881がアイドル状態であることを表す
cirread	応答レジスタの再度読み出し
cirbusy	68881のBUSYを表す

実行アドレス評価
データ転送プリミティブの内容

●OPクラス010,011のもの

op010_b	バイト整数
op010_w	ワード整数
op010l_s	ロングワード整数/単精度実数
op010_d	倍精度実数
op010_xp	拡張精度実数/パック型10進

●レジスタ群

fp0-fp7	数値演算プロセッサレジスタ
fpsr,fpcr	数値演算プロセッサコントロールレジスタ

●ソースデータフォーマット

_BYTE	バイト
_WORD	ワード
_LONG	ロングワード
_SINGLE	シングルリアル(単精度実数)
_DOUBLE	ダブルリアル（倍精度実数)
_EXTENDED	エクステンドリアル(拡張精度実数)
_PACKED	パックドデシマルリアル（パック型10進)

●ROM定数オフセット値

MOVECR命令で使用するもの。

pi	π
log10_2	$\log_{10}2$
exp	e:自然指数
log2_e	$\log_2 e$
log10_e	$\log_{10}(e)$
zero	0.0
ln_2	$\ln(2)=\log_e 2$
ln_10	$\ln(2)=\log_e 10$
exp0	10^0
exp1	10^1
exp2	10^2
exp4	10^4
exp8	10^8
exp16	10^{16}
exp32	10^{32}
exp64	10^{64}
exp128	10^{128}
exp256	10^{256}
exp512	10^{512}
exp1024	10^{1024}
exp2048	10^{2048}
exp4096	10^{4096}

●FBcc,FDBcc命令のコンディションコード

_F_F	偽
_F_EQ	等しい
_F_OGT	オーダーでより大きい
_F_OGE	オーダーでより大きいか等しい
_F_OLT	オーダーでより小さい
_F_OLE	オーダーでより小さいか等しい
_F_OGL	オーダーでより大きいか，小さい
_F_OR	オーダー
_F_UN	アンオーダー
_F_UEQ	アンオーダーか等しい
_F_UGT	アンオーダーかより大きい
_F_UGE	アンオーダーかより大きいか等しい
_F_ULT	アンオーダーかより小さい
_F_ULE	アンオーダーかより小さいか等しい
_F_NE	等しくない
_F_T	真

以後の条件は，68881内部のNAN(Not A Number:数でないもの)のビットがセットされていると，FPSRレジスタのBUNがセットされる。

_F_SF	シグナリング，偽
_F_SEQ	シグナリング，等しい
_F_GT	より大きい
_F_GE	より大きいか等しい
_F_LT	より小さい
_F_LE	より小さいか等しい
_F_GL	より大きいかより小さい
_F_GLE	より大きいかより小さいか等しい
_F_NGLE	GLEでない
_F_NGL	GLでない
_F_NLE	LEでない
_F_NLT	LTでない

_F_NGE　GEでない
_F_NGT　GTでない
_F_SNE　シグナリング，等しくない
_F_ST　シグナリング，真

マクロ命令

まず，マクロ中に利用されるマクロで，ユーザープログラム中ではほぼ使われないものを紹介する（outreg：出力データ，cirreg：CIRレジスタ）。

●書き込み命令

MOVE.Wと変わらないが，シンボル（A5reg）をチェックする。それぞれ，バイト，ワード，ロングワード型。

```
CIROUTB  outdata,cirreg
CIROUTW  outdata,cirreg
CIROUTL  outdata,cirreg
```

●ロングワード長書き込み命令

```
CIRINB   outdata,cirreg
CIRINW   outdata,cirreg
CIROUTL  outdata,cirreg
```

例）

```
CIROUTW sfppreg<<10+dfppreg<<7+op,CIRCMD
CIROUTL sl, CIROPR
```

●応答プリミティブ用ウェイト命令

応答CIRがコンディションの値になるまで待つ（cond：コンディション）。

```
W_STAT  cond
```

例）

```
W_STAT  cirnull
```

●応答CIRのカムアゲインビットのチェック

カムアゲインビットが寝ていたら真。

```
CMCHK
```

次に，プログラム中で使用されるマクロについて説明する。

●FSTART

数値演算プロセッサの初期化をするため，用意したもので，68881の例外状態からの復帰を行う（リセットに意味が近い）。

●OP000　sfppreg,dfppreg,op

●OP010　s1，s2，s3，dfooreg,op,sdf

sfppreg ソースFPPレジスタ
dfppreg デスティネーションFPPレジスタ
op　命令コード
sl　外部ソース1（.x,.pのみ利用）
s2　外部ソース2（.d,.x,.pのみ使用）
s3　外部ソース3（すべて利用）
sdf ソースデータフォーマット（下記参照）

OP000はFPPレジスタ間転送命令やFPPレジスタ間演算命令などを行う。dfppregとsfppregのレジスタを同じにすると単項演算命令になる。ソースフォーマットは拡張倍精度に固定。

OP010は外部→レジスタ間のデータ転送命令で，ソースレジスタを68000側のポインタなどを使った表記や，データレジスタを使った表記が可能。なお，sl，s2，s3の関係は，

$$sl\times 2^{64}+s2\times 2^{32}+s3$$

である。デスティネーションが拡張倍精度もしくはパック型10進ならsl，s2，s3を使用する。倍精度ならs2，s3を使用し，slはダミー。それ以下のビットのソースフォーマットは，s3だけ利用しsl，s2はダミーとなる。

なお，数値演算プロセッサ内のレジスタFP0〜FP7は，どのような型から代入されても，拡張精度実数に変換される。データの丸めも同時に行われるので注意。

●FPP.MACで使用可能な演算命令

FABS　|sfppreg|（絶対値）→dfppreg
FACOS　arccos(sfppreg)→dfppreg
*FADD　sfppreg＋dfppreg→dfppreg
FASIN　arcsin(sfppreg)→dfppreg
FATAN　arctan(sfppreg)→dfppreg
FATANH　arctanh(sfppreg)→dfppreg
*FCMP　dfppreg-sfppreg（比較のみ）
FCOS　cos(sfppreg)→dfppreg
FCOSH　cosh(sfppreg)→dfppreg
*FDIV　dfppreg÷sfppreg→dfppreg
FETOX　$e^{sfppreg}$→dfppreg
FETOXM1　$e^{sfppreg}-1$→dfppreg
FGETEXP　$s^{sfppreg}$の指数部→dfppreg
FGETMAN　sfppregの仮数部→dfppreg
FINT　sfppregの整数部→dfppreg
FINTRZ　sfppregの整数化→dfppreg
FLOG10　$\log_{10}$(sfppreg)→dfppreg
FLOG2　$\log_{02}$(sfppreg)→dfppreg
FLOGN　ln(sfppreg)→dfppreg
FLOGNP1　ln(sfppreg＋1)の→dfppreg
*FMOD　(dfppreg)mod(sfppreg)→dfppreg
*FMOVE　sfppreg→dfppreg
*FMUL　sfppreg×dfppreg→dfppreg
*FNEG　−(sfppreg)→dfppreg
*FREM　dfppreg÷sfppreg(IEEE除算)→dfppreg
*FSCALE　dfppreg×int($2^{sfppreg}$)→dfppreg
*FSGLDIV　dfppreg÷sfppreg(単精度)→dfppreg
*FSGLMUL　sfppreg×dfppreg(単精度)→dfppreg
FSIN　sin(sfppreg)→dfppreg
FSINH　sinh(sfppreg)→dfppreg
FSQRT　sfppreg→dfppreg
*FSUB　dfppreg-sfppreg→dfppreg
FTAN　tan(sfppreg)→dfppreg
FTANH　tanh(sfppreg)→dfppreg
FTENTOX　$10^{sfppreg}$→dfppreg
FTST　sfppregのテスト（単項演算のみ）
FTWOTOX　$2^{sfppreg}$→dfppreg

*のついた命令は単項演算ができない命令。なお，ln（Natural Log)は$\log_e$のことである。

例）

```
OP000 fp0, fp1, FADD
OP000 fp0,  fp0,  SINH     * 単項演算
OP010 d0, d1, d2, fp0, MOVE, _EXTENDED
OP010 #0,  #0,  d0,  fp0,  MOVE,  _LONG
OP010 d0, d1, d2, fp0, FSIN, _EXTENDED
```

●FSINCOS命令

この命令は特殊で，デスティネーションが2つあるものである。したがって通常の表記ではできないため以下のように行う。ちなみに，1命令の時間でSIN関数，COS関数の2つの計算を同時に行うため，非常に速い。

例）　OP000 fp0, fp1, FSINCOS＋fp2
　式）sin(fp0)=fp1
　　cos(fp0)=fp2

●OP011 sfppreg,d1，d2，d3，Kfactor，ddf

sfppreg　ソースFPPレジスタ
d1　外部デスティネーション1（.x,.pのみ利用）
d2　外部デスティネーション2（.d,.x,.pのみ使用）
d3　外部デスティネーション3（すべて利用）
Kfactor　Kファクター値（ただし静的）
ddf　デスティネーションデータフォーマット

これはFPPレジスタ→外部転送専用fmove命令。d1，d2，d3の使い方は，前項のs1，s2，s3とほぼ同様。

Kfactorは，送り先のデータがパック型10進の場合のみ必要。そのほかではすべて0にすること。このKFactorの値は2の補数の整数でコード化しなくてはならない。この値は7ビットの整数で次のような意味をもつ。

−64〜0　10進小数点の右側の有効数字の数
1〜17　仮数内有効数字数
18〜63　FPSR例外バイト内でOPERRビットがセットされ＋17として扱われる

なお，動的Kファクターの設定には対応していない。−64〜−1は必ず4ビット16進に変換すること。KFactorのデータフォーマットは，デスティネーションのデータフォーマットと同じなので注意。外部に転送されるときは，自動的にそれに適した丸めが行われる。

例）

```
OP011 fp0,  d0,  d1,  d2,  0,  _EXTENDED
OP011 fp0,  d0,  d1,  d2,  +5,  _PACKED
```

●FMOVECR ofs,dfppreg

ofs　定数ROMオフセット値
dfppreg デスティネーションFPPレジスタ

ROM定数内容転送命令。MC68881のオンチップROMから拡張精度定数を転送し，FPCRにより指定される精度まで丸めてdfppregに転送する。オフセット値は，ROM定数オフセット値の項参照のこと。

例）　MOVECR pi,FP0

●OP100 sfppcrreg,d1

●OP101 s1，dfppcrreg

sfppcrreg　FPCR MC68881内部コントロールレジスタ
dfppcrreg　FPCR MC68881内部コントロールレジスタ
d1　デスティネーション値。
s1　ソース値

浮動小数点システムコントロールレジスタの内容を直接やりとりする。ソースデータフォーマットはlongに固定。

FPCRはMC68881で起きる例外的なイネーブル，ディスイネーブルの制御，丸め精度の変更を行い，FPSRは分岐命令のためのフラグ，割り算の商ビットの格納を行う。

例）

```
OP100 FPCR,d0
OP101 d0, FPSR
```

●FBcc　cc,braadr

cc　条件（コンディションコード）
braadr　ブランチアドレス

条件を満たしている場合は分岐を行う。Bcc命令のFPPレジスタ対応版。

例）　FBcc _FGT,braadr

●FDBcc cc,dreg,braadr

cc　条件（コンディションコード）
dreg　データレジスタ（d0〜d7）
braadr　ブランチアドレス

条件が真の場合は，演算を行わずにデータレジスタのデクリメントを行う。このとき，データレジスタが−1でない場合，braadrに分岐する。

例）　FDBcc _FGT，d0，braadr

リスト1 FPP.MAC

```
        .nlist
*
*  数値演算プロセッサ（ｍｃ６８８８１）用マクロ
*
*
*       以下の文字列を定義することにより、
*       ２種の動作に分岐します
*
*       A5reg   定義    Ａ５相対アドレッシングモード
*                       はやいがＡ５レジスタがつかえない
*               非定義  直接アドレッシング
*                       遅いがすべてのレジスタがつかえる
*
* 初期化
*
FSTART  macro                   * Ｆｐｐアドレスの転送
        ifdef A5reg
        lea     CIRADR,A5
        endif
        CIROUTW #$0001,CIRCON   * Fpp アボート
        endm
*
* ＣＩＲ 書き込み命令
*
CIROUTB macro   outdata,cirreg  * ＣＩＲ  アウト（バイト）命令
        ifdef   A5reg
        move.b  outdata,cirreg(a5)
        else
        move.b  outdata,cirreg+CIRADR
        endif
        endm
CIROUTW macro   outdata,cirreg  * ＣＩＲ  アウト（ワード）命令
        ifdef   A5reg
        move.w  outdata,cirreg(a5)
        else
        move.w  outdata,cirreg+CIRADR
        endif
        endm
CIROUTL macro   outdata,cirreg  * ＣＩＲ  アウト（ロングワード）命令
        ifdef   A5reg
        move.l  outdata,cirreg(a5)
        else
        move.l  outdata,cirreg+CIRADR
        endif
        endm
*
* ＣＩＲ 読み込み命令
*
CIRINB  macro   cirreg,indata   * ＣＩＲ  イン（バイト）命令
        ifdef   A5reg
        move.b  cirreg(a5),indata
        else
        move.b  cirreg+CIRADR,indata
        endif
        endm
CIRINW  macro   cirreg,indata   * ＣＩＲ  イン（ワード）命令
        ifdef   A5reg
        move.w  cirreg(a5),indata
        else
        move.w  cirreg+CIRADR,indata
        endif
        endm
CIRINL  macro   cirreg,indata   * ＣＩＲ  イン（ロングワード）命令
        ifdef   A5reg
        move.l  cirreg(a5),indata
        else
        move.l  cirreg+CIRADR,indata
        endif
        endm
*
*  応答プリミティブ用ウエイト命令
*
W_STAT  macro   cond
        local   w_loop
w_loop: ifdef A5reg
        cmpi.w  #cond,CIRRES(a5)
        else
        cmpi.w  #cond,CIRRES+CIRADR
        endif
        bne     cirerr
        bra     w_stat_end
        local   cirerr
cirerr:
        ifdef A5reg
        cmpi.b  #$1d,CIRRES(a5)
        else
        cmpi.b  #$1d,CIRRES+CIRADR
        endif
        bne     w_loop
        CIROUTW #$0001,CIRCON           * 例外からの解除
        local   w_stat_end
w_stat_end:
        endm
*W_STAT macro   cond
*       local   w_loop
*w_loop:
*       ifdef A5reg
*       cmpi.w  #cond,CIRRES(a5)
*       else
*       cmpi.w  #cond,CIRRES+CIRADR
*       endif
*       bne     w_loop
*       endm
*  カムアゲインビットチェック
*
CMCHK   macro
        ifdef A5reg
        btst.b  #7,CIRRES(a5)
        else
        btst.b  #7,CIRRES+CIRADR
        endif
        endm
*
* 一般的な命令  ＯＰクラス０００  レジスターレジスタ間
*
OP000   macro   sfppreg,dfppreg,op
        * sfppreg : ソースレジスタ
        * dfppreg : ディスティネーションレジスタ
        * op      : 命令コード
        W_STAT  cirnull
        CIROUTW #(sfppreg<<10)+(dfppreg<<7)+op,CIRCMD
        endm
*
* 一般的な命令  ＯＰクラス０１０  外部ーレジスタ間
*
OP010   macro   s1,s2,s3,dfppreg,op,sdf
        * s1      : 外部ソース１（.x,.pのみ利用）
        * s2      : 外部ソース２（.d,.x,.pのみ使用）
        * s3      : 外部ソース３（すべて利用）
        * sdf     : ソースデータフォーマット（下記参照）
        W_STAT  cirnull
        CIROUTW #$4000+(sdf<<10)+(dfppreg<<7)+op,CIRCMD
        if sdf=_BYTE
                W_STAT  op010_b  * バイト
                W_STAT  cirread
                CIROUTB s3,CIROPR
        elseif sdf=_WORD
                W_STAT  op010_w  * ワード
                W_STAT  cirread
                CIROUTW s3,CIROPR
        elseif (sdf=_LONG).or.(sdf=_SINGLE)
                W_STAT  op010_ls * ロングワード／単精度実数
                W_STAT  cirread
                CIROUTL s3,CIROPR
        elseif (sdf=_DOUBLE)
                W_STAT  op010_d  * 倍精度実数
                W_STAT  cirread
                CIROUTL s2,CIROPR
                CIROUTL s3,CIROPR
        elseif (sdf=_EXTENDED).or.(sdf=_PACKED)
                W_STAT  op010_xp * 拡張精度実数／パック型10進
                W_STAT  cirread
                CIROUTL s1,CIROPR
                CIROUTL s2,CIROPR
                CIROUTL s3,CIROPR
        endif
*       W_STAT  cirbusy
        endm
FMOVECR macro   ofs,dfppreg
        * MC68881内部定数ＲＯＭの読み出す
        * フォーマットは拡張精度のみ
        * ofs     : オフセット（内容は下記参照）
        W_STAT  cirnull
        CIROUTW #$5c00+(dfppreg<<7)+ofs,CIRCMD
        endm
OP011   macro   sfppreg,dst1,dst2,dst3,Kfactor,ddf
        * dst1      : 外部ディスティネーション１（.x,.pのみ利用）
        * dst2      : 外部ディスティネーション２（.d,.x,.pのみ使用）
        * dst3      : 外部ディスティネーション３（すべて利用）
        * Kfactor : Ｋファクター値
        W_STAT  cirnull
        CIROUTW #$6000+(ddf<<10)+(sfppreg<<7)+Kfactor,CIRCMD
        if ddf=_BYTE
                W_STAT  op011_b
                CIRINB  CIROPR,dst3
        elseif ddf=_WORD
                W_STAT  op011_w
                CIRINW  CIROPR,dst3
        elseif (ddf=_LONG).or.(ddf=_SINGLE)
                W_STAT  op011_ls
                CIRINL  CIROPR,dst3
        elseif ddf=_DOUBLE
                W_STAT  op011_d
                CIRINL  CIROPR,dst2
                CIRINL  CIROPR,dst3
        elseif (ddf=_EXTENDED).or.(ddf=_PACKED)
                W_STAT  op011_xp
                CIRINL  CIROPR,dst1
                CIRINL  CIROPR,dst2
                CIRINL  CIROPR,dst3
        endif
        endm
OP100   macro   sfppcrreg,dst1
        * sfppcrreg:FPcr MC68881内部コントロールレジスタ
        W_STAT  cirnull
        CIROUTW #$a000+(sfppcrreg<<10),CIRCMD
        W_STAT  cirread
        CIRINL  CIROPR,dst1
        endm
OP101   macro   s1,dfppcrreg
        * dfppcrreg:FPcr MC68881内部コントロールレジスタ
        W_STAT  cirnull
        CIROUTW #$8000+(dfppcrreg<<10),CIRCMD
        W_STAT  cirread
        CIROUTL s1,CIROPR
        endm
FBcc    macro   cc,braadr
        * cc      : 条件（下記参照）
        * braadr  : ブランチアドレス
        W_STAT  cirnull
        CIROUTW #cc,CIRCMD
        ifdef A5reg
        cmp.w   #circc,CIRRES(a5)
        else
        cmp.w   #circc,CIRRES+CIRADR
        endif
        beq     braadr
        endm
FDBcc   macro   cc,dreg,braadr  * dreg:データレジスタ
        W_STAT  cirnull
        CIROUTW #cc,CIRCMD
        ifdef A5reg
        cmp.w   #circc,CIRRES(a5)
        else
        cmp.w   #circc,CIRRES+CIRADR
        endif
        dbeq    dreg,braadr
        endm
*
*       define device address
*
*                               Address * CIR
CIRADR          equ     $E9E000 * Base Address
CIRRES          equ     $00     * Responce
CIRCON          equ     $02     * Contorol
CIRSAVE         equ     $04     * Save
CIRREST         equ     $06     * Restore
CIRCMD          equ     $0A     * Command
CIRCND          equ     $0E     * Condition
CIROPR          equ     $10     * Operand(Data)
CIRRS           equ     $14     * Register Select
*
```

```
*
*       応答CIRの内容          (ヌルプリミティブ)
*
*                               Code    * Status
circc           equ     $0800   * コンディションCIRへの
                                * 書き込みに対する応答(真)
cirnull         equ     $0802   * 68881がアイドル状態であることを表わす。
cirread         equ     $8900   * 応答レジスタの再度読み出し
cirbusy         equ     $0900   * 68881がBUSYである場合
*
*       実行アドレス評価/データ転送プリミティブの内容
*
op010_b         equ     $9501
op010_w         equ     $9502
op010_ls        equ     $9504
op010_d         equ     $9508
op010_xp        equ     $960c

op011_b         equ     $b101
op011_w         equ     $b102
op011_ls        equ     $b104
op011_d         equ     $b208
op011_xp        equ     $b20c

op100_4         equ     $9504
op100_8         equ     $9608
op100_12        equ     $960c

op101_4         equ     $b104
op101_8         equ     $b208
op101_12        equ     $b20c


*
*       define fpp register name
*

fp0             equ     0
fp1             equ     1
fp2             equ     2
fp3             equ     3
fp4             equ     4
fp5             equ     5
fp6             equ     6
fp7             equ     7
fpsr            equ     2
fpcr            equ     4
*
*       S)ource D)ata F)ormat
*
_LONG           equ     0
_SINGLE         equ     1
_EXTENDED       equ     2
_PACKED         equ     3
_WORD           equ     4
_DOUBLE         equ     5
_BYTE           equ     6
*
*       define function code
*
FABS            equ     $18
FACOS           equ     $1C
FADD            equ     $22
FASIN           equ     $0C
FATAN           equ     $0A
FATANH          equ     $0D
FCMP            equ     $38
FCOS            equ     $1D
FCOSH           equ     $19
FDIV            equ     $20
FETOX           equ     $10
FETOXM1         equ     $08
FGETEXP         equ     $1E
FGETMAN         equ     $1F
FINT            equ     $01
FINTRZ          equ     $03
FLOG10          equ     $15
FLOG2           equ     $16
FLOGN           equ     $14
FLOGNP1         equ     $06
FMOD            equ     $21
FMOVE           equ     $00
FMUL            equ     $23
FNEG            equ     $32
FREM            equ     $25
FSCALE          equ     $26
FSGLDIV         equ     $24
FSGLMUL         equ     $27
FSIN            equ     $0E
FSINCOS         equ     $30
FSINH           equ     $02
FSQRT           equ     $04
FSUB            equ     $28
FTAN            equ     $0F
FTANH           equ     $09
FTENTOX         equ     $12
FTST            equ     $3A
FTWOTOX         equ     $11
*
*       ROM定数オフセット
*
pi              equu    $00             * π
log10_2         equ     $0B             * log10(2)
exp             equ     $0c             * e:自然指数
log2_e          equ     $0d             * log 2(e)
log10_e         equ     $0e             * log10(e)
zero            equ     $0f             * 0.0
ln_2            equ     $30             * ln(2)=log e( 2)
ln_10           equ     $31             * ln(2)=log e(10)
exp0            equ     $32             * 10^0
exp1            equ     $33             * 10^1
exp2            equ     $34             * 10^2
exp4            equ     $35             * 10^4
exp8            equ     $36             * 10^8
exp16           equ     $37             * 10^16
exp32           equ     $38             * 10^32
exp64           equ     $39             * 10^64
exp128          equ     $3a             * 10^128
exp256          equ     $3b             * 10^256
exp512          equ     $3c             * 10^512
exp1024         equ     $3d             * 10^1024
exp2048         equ     $3e             * 10^2048
exp4096         equ     $3f             * 10^4096
*
*       比較
*
_F_F            equ     $00
_F_EQ           equ     $01
_F_OGT          equ     $02
_F_OGE          equ     $03
_F_OLT          equ     $04
_F_OLE          equ     $05
_F_OGL          equ     $06
_F_OR           equ     $07
_F_UN           equ     $08
_F_UEQ          equ     $09
_F_UGT          equ     $0a
_F_UGE          equ     $0b
_F_ULT          equ     $0c
_F_ULE          equ     $0d
_F_NE           equ     $0e
_F_T            equ     $0f
_F_SF           equ     $10
_F_SEQ          equ     $11
_F_GT           equ     $12
_F_GE           equ     $13
_F_LT           equ     $14
_F_LE           equ     $15
_F_GL           equ     $16
_F_GLE          equ     $17
_F_NGLE         equ     $18
_F_NGL          equ     $19
_F_NLE          equ     $1a
_F_NLT          equ     $1b
_F_NGE          equ     $1c
_F_NGT          equ     $1d
_F_SNE          equ     $1e
_F_ST           equ     $1f

        .list
```

リスト2　FSIN.S

```
        .include fpp.mac
        .include doscall.mac

        .xdef   _fsin


        .offset 4
num     ds.l    1

        .text

_fsin:  move.l  num(a7),d2

_mc881: clr.l   -(SP)                           * スーパバイザモードに移行
        DOS     _SUPER
        addq.l  #4,SP
        move.l  d0,_SSP
        FSTART
        OP010   #0,#0,d2,fp0,FSIN,_SINGLE       * fp0=sin(d2)
fsin:   OP011   fp0,#0,#0,d2,0,_SINGLE
fmove:  move.l  _SSP,-(SP)              * ユーザモードに復帰
        DOS     _SUPER
        addq.l  #4,SP
        move.l  d2,d0
        rts

        .bss
_SSP:
        ds.l    1
        .end
```

リスト3　FFT.S

```
        .include        fpp.mac
        .include        doscall.mac

*A5reg  equ

        .xdef   _fft


        .offset 4
CMPLX_y ds.l    1
CMPLX_c ds.l    1
num     ds.l    1
iw      ds.l    1

savesize        equ     4*9

        .text

_fft:
        movem.l d3-d7/a3-a6,-(sp)
        movem.l CMPLX_y+savesize(a7),a5-a6
        movem.l num+savesize(a7),d6-d7

_strt:  move.l  a6,a1           * c
        move.l  a5,a0           * y
        move.l  d6,d0           * num
        add.l   #8,a0
        add.l   #8,a1

fr_i:
        move.l  (a0)+,(a1)+     * c[i].Re = y[i].Re
        move.l  (a0)+,(a1)+     * c[i].Im = y[i].Im
        subq.l  #1,d0
        bne     fr_i

        move.l  #1,d3           * c[i] の i(d3)
        move.l  #1,d4           * c[j] の j(d4)
fr_j_i:
        cmp.l   d3,d6           * num(d6)-i(d3)<0
        blt     _mc881
        cmp.l   d3,d4           * j(d4)=i(d3)=<0
```

▶プログラミングに何度も入門しようとしている私にとって，7月号の特集はとても役に立ちました。年に一度くらいは入門したいです。　春名　義行(25)兵庫県

```
        ble     m_n
i_j:
        lsl.l   #3,d4                   * *8
        lsl.l   #3,d3                   * *8
        move.l  (a6,d3.l),d0            * c[i].Re=>d0
        move.l  (a6,d4.l),(a6,d3.l)     * c[j].Re=>c[i].Re
        move.l  d0,(a6,d4.l)            * d0=>c[j].Re
        move.l  4(a6,d3.l),d0           * c[i].Im=>d0
        move.l  4(a6,d4.l),4(a6,d3.l)   * c[j].Im=>c[i].Im
        move.l  d0,4(a6,d4.l)           * d0=>c[j].Im
        lsr.l   #3,d3                   * 8
        lsr.l   #3,d4
m_n:
        move.l  d6,d5                   * n(d6)=>m(d5)
        asr.l   #1,d5                   * m(d5)/2=>m(d5)
while_j:
        cmp.l   d5,d4                   * j(d4)-m(d5)<=0
        ble     wend_j
        sub.l   d5,d4                   * j(d4)=j-m(d5)
        asr.l   #1,d5                   * m(d5)/2=>m(d5)
        cmp.l   #2,d5                   * m(d5)-2>=0
        bge     while_j
wend_j:
        add.l   d5,d4                   * j(d4)=j(d4)+m(d5)
        addq.l  #1,d3
        bra     fr_j_i

_mc881: clr.l   -(SP)                   * スーパバイザモードに以降。
        DOS     _SUPER
        addq.l  #4,SP
        move.l  d0,_SSP
        FSTART

        move.l  #1,d5                   * 1=>max(d5)
while_max:
        cmp.l   d5,d6                   * num(d6)-max(d5)
        ble     wend_max
        move.l  d5,d2                   * max(d5)=>is
        add.l   d2,d2                   * is(d2)*2=>is
        move.l  #1,d1                   * 1=>k(d1)
fr_k:
        cmp.l   d1,d5                   * max(d5)-k(d1)<0
        blt     fr_k_end

        FMOVECR pi,fp7
        OP010   #0,#0,d7,fp0,FMOVE,_LONG * iw(fp0)=iw(d7)
        OP000   fp7,fp0,FMUL             * iw(fp0)=iw*pi(fp7)
        subq.l  #1,d1
        OP010   #0,#0,d1,fp0,FMUL,_LONG  * iw(fp0)=iw*(k-1)(d1)
        addq.l  #1,d1
        OP010   #0,#0,d5,fp0,FDIV,_LONG  * theta(fp0)=iw/max(d5)

        move.l  d1,d3                   * i(d3)=k(d1)
_fsc0:  OP000   fp0,fp2,FSINCOS+fp1     * fp2=sin(theta) fp1=cos(theta)

fr_i2:
        cmp.l   d3,d6                   * n(d6)-i(d3)<0
        blt     fr_i2_end
        move.l  d3,d4                   * j(d4)=i(d3)
        add.l   d5,d4                   * j=j+max
        lsl.l   #3,d4                   * *8
        lsl.l   #3,d3                   * *8
        move.l  (a6,d4.l),d0
        OP010   #0,#0,d0,fp4,FMOVE,_SINGLE      * c[j].Re(fp4,fp0)
        OP000   fp4,fp0,FMOVE
        move.l  4(a6,d4.l),d0
        OP010   #0,#0,d0,fp5,FMOVE,_SINGLE      * c[j].Im(fp5,fp3)
        OP000   fp5,fp3,FMOVE

L1      OP000   fp1,fp0,FMUL    * fp0=c[j].Re(fp0) * cos(fp1)
        OP000   fp2,fp3,FMUL    * fp3=c[j].Im(fp3) * sin(fp2)
        OP000   fp3,fp0,FSUB            * (fp0)str.Re

L2      OP000   fp1,fp5,FMUL    * fp5=c[j].Im(fp5) * cos(fp1)
        OP000   fp2,fp4,FMUL    * fp4=c[j].Re(fp4) * sin(fp2)
        OP000   fp5,fp4,FADD            * (fp4)str.Im

        move.l  (a6,d3.l),d0
L3      OP010   #0,#0,d0,fp3,FMOVE,_SINGLE      * fp3=c[i].Re
        OP000   fp3,fp5,FMOVE                   * fp5=c[i].Re
        move.l  4(a6,d3.l),d0
        OP010   #0,#0,d0,fp6,FMOVE,_SINGLE      * fp6=c[i].Im
        OP000   fp6,fp7,FMOVE                   * fp7=c[i].Im

L4      OP000   fp0,fp3,FSUB    * c[j].Re(fp3)=c[i].Re(fp3)-str.Re(fp0)
        OP000   fp4,fp6,FSUB    * c[j].Im(fp6)=c[i].Im(fp6)-str.Im(fp4)
        OP000   fp0,fp5,FADD    * c[i].Re(fp5)=c[i].Re(fp5)+str.Re(fp0)
        OP000   fp4,fp7,FADD    * c[i].Im(fp7)=c[i].Im(fp7)+str.Im(fp4)

L5
        OP011   fp3,#0,#0,d0,0,_SINGLE  * c[j].Re(fp3)
        move.l  d0,(a6,d4.l)
        OP011   fp6,#0,#0,d0,0,_SINGLE  * c[j].Im(fp6)
        move.l  d0,4(a6,d4.l)
        OP011   fp5,#0,#0,d0,0,_SINGLE  * c[i].Re(fp5)
        move.l  d0,(a6,d3.l)
        OP011   fp7,#0,#0,d0,0,_SINGLE  * c[i].Im(fp7)
        move.l  d0,4(a6,d3.l)

        lsr.l   #3,d4
        lsr.l   #3,d3

        add.l   d2,d3                   * i=i+is
        bra     fr_i2
fr_i2_end:
        addq.l  #1,d1
        bra     fr_k
fr_k_end:
        move.l  d2,d5                   * is(d2)=>max(d5)
        bra     while_max
wend_max:
        cmp.l   #1,d7                   * iw(d7)-1=0
        beq     fftend
        moveq.l #1,d3                   * i(d3)=1
fr_i3:
        cmp.l   d3,d6                   * num(d6)-i(d3)<0
        blt     fftend
        lsl.l   #3,d3                   * *8
        move.l  (a6,d3.l),d0
        OP010   #0,#0,d0,fp0,FMOVE,_SINGLE * fp0=c[i].Re
        OP010   #0,#0,d6,fp0,FDIV,_LONG * c[i].Re=c[i].Re(fp0)/num(d6)
        OP011   fp0,#0,#0,d0,0,_SINGLE
        move.l  d0,(a6,d3.l)
        move.l  4(a6,d3.l),d0
        OP010   #0,#0,d0,fp0,FMOVE,_SINGLE * fp0=c[i].Im
        OP010   #0,#0,d6,fp0,FDIV,_LONG * c[i].Im=c[i].Im(fp0)/num(d6)
        OP011   fp0,#0,#0,d0,0,_SINGLE
        move.l  d0,4(a6,d3.l)
        lsr.l   #3,d3
        addq    #1,d3
        bra     fr_i3
fftend:
        move.l  _SSP,-(SP)      * ユーザモードに復帰
        DOS     _SUPER
        addq.l  #4,SP
        movem.l (sp)+,d3-d7/a3-a6
        rts

        .bss
_SSP:
        ds.l    1

        .end
```

リスト4 MAINF.C

```
#include        <basic0.h>
#include        <basic.h>
#include        <mouse.h>
#include        <graph.h>
#include        <fcntl.h>
#include        <math.h>

typedef struct cmplx {  /*  複素数型 */
        float   Re;
        float   Im;
} CMPLX;


#define NASI    0x4e415349      /* 'NASI' */
#define BUFF_SIZE       512

        CMPLX   c[BUFF_SIZE+1],
                y[BUFF_SIZE+1],
                z[BUFF_SIZE+1];

        double  s[64+1];
        short   data[BUFF_SIZE+1];
        short   moto[BUFF_SIZE+1];

        char    fileName[40];
        int     fn;

void    drawGraph(short *,int);
void    fft(CMPLX *,CMPLX *, int, int);


main() {
unsigned char   strtmp0[258];
unsigned char   key[1];

        int     i;
        int     ms_x,ms_y,ms_bl,ms_br;

        console(0,32,0);
        initScreen();
        for (i=0;i<=511;i++){
                data[i]=0;
        }

        box(30,30,30+512,30+256,8,NASI);
        line(30,30+128,30+512,30+128,8,NASI);
        mouse(4);
        mouse(1);
        msarea(6,300,699,428);

        while (1) {
             msstat(&ms_x,&ms_y,&ms_bl,&ms_br);
             mspos(&ms_x,&ms_y);
             if(ms_bl == -1) mslDown(ms_x, ms_y);
             if(ms_br == -1) msrDown(ms_x, ms_y);

/*リアルタイム一文字入力*/
             strncpy(key, b_inkey0(strtmp0), sizeof(key));

             switch(toupper(*key)) {
                  case 'A':
                          analize();
                          break;
                  case 'E':
                          mouse(0);
                          wipe();
                          exit(0);
                          break;
                  case 'V':
                          view();
                          break;
                  case 'L':
                          loadData();
                          break;
                  case 'S':
                          saveData();
                          break;
             }

        }
}

clrWin(int line) {
        int     cnt;

        locate(0,28);
        while(line--) {
                cnt = 60;
                while(cnt--) putchar(' ');
                putchar('¥n');
        }
}

/*
```

▶文月さんの副業は，自動車会社の仕事だったんですね。本業はライターですか？
新子　弘康(21)大阪府

```c
/*
**  マウスの左ボタンが押された
*/
mslDown(int x, int y) {
        int     ch;

        ch=(x - 3) / 11;
        erasVr(ch);
        if(s[ch] >= 0) {
                s[ch]=(double)(428-y) / 30.0;
                s[ch]=pow((double)10.0,s[ch]);
        } else {
                s[ch]=(double)(428-y) / 30.0;
                s[ch]=-pow((double)10.0,s[ch]);
        }
        drawVr(ch);
}

/*
**  マウスの右ボタンが押された
*/
msrDown(int x,int y) {
        int     ch;
        int     tmp;

        ch=(x - 3) / 11;
        locate(0,28);
        if ( x < 353 ) {
           printf("ＳＩＮ関数の%d番目の係数の値は%4.2fです        ¥n",
           ch+1,s[ch]);
           } else {
           printf("ＣＯＳ関数の%d番目の係数の値は%4.2fです        ¥n",
                        ch-31,s[ch]);
        }
        b_input("新しい値を入れて下さい－>",0x204,&tmp,-1);
        erasVr(ch);
        s[ch]=tmp % 0x8000;
        drawVr(ch);
        clrWin(2);
}

/*
**  画面の初期化
*/
int     initScreen() {
        int     i;

        screen(2,0,1,1);
        for (i=0;i<=31;i++){
                line(6+i*11,300,6+i*11,428,9,NASI);
                line(358+i*11,300,358+i*11,428,9,NASI);
        }
        line(351,300,351,428,5,NASI);
        locate(76,3);
        puts("Commands ");
        locate(77,5);
        puts("A)nalize");
        locate(77,7);
        puts("V)iew ");
        locate(77,9);
        puts("S)ave ");
        locate(77,11);
        puts("L)oad ");
        locate(77,13);
        puts("E)xit ");

        for (i=0;i<=63;i++){    /*ボリュームを描く*/
                s[i]=0;
                drawVr(i);
        }
}

/*
**  ボリュームを描く
*/
int     drawVr(int ch) {

        int     x,y;

        x = ch*11;
        if(s[ch] == 0)
                y = 428;
        else
                y = 428-log10(fabs(s[ch]))*30;

        if( y < 300) y = 300;
        if( y > 428) y = 428;
        box(x,y,x+11,y,9,NASI);
}

/*
**  ボリュームを消す
*/
int     erasVr( int ch) {

        int     x,y;

        x = ch*11;
        if(s[ch] == 0)
                y = 428;
        else
                y = 428-log10(fabs(s[ch]))*30;

        if( y < 300) y = 300;
        if( y > 428) y = 428;

        fill(x,y,x+11,y,0);

        line(6+ch*11,300,6+ch*11,428,9,NASI);
        if ( ch==31 | ch==32 ) {
                line(351,300,351,428,5,NASI);
        }
}

/*
**  画面に波形を描く
*/
void    drawGraph(short dat[], int clr) {
        int     oy,ny;
        int     i;

        oy=158-dat[0]/32;
        for (i=1; i <= BUFF_SIZE ; i++) {
                ny = 158 - dat[i] / 32;
                if(ny > 30+256) ny = 30+256;
                if(ny < 30)     ny = 30;
                line(i+29,oy,i+30,ny,clr,NASI);
                oy=ny;
        }
        box(30,30,30+512,30+256,8,NASI);
        line(30,30+128,30+512,30+128,8,NASI);
}

/*
**      ファイルより読み込む
*/
loadData() {
        int     fn;
        int     i;

        int     length;

        locate(0,28);
        b_input("File Name: ",sizeof(fileName),fileName,-1);
        clrWin(1);

        fn = open(fileName, O_BINARY | O_RDONLY);
        if (fn >= 0) {
                length = filelength(fn);
                if (length > BUFF_SIZE * sizeof(short)) {
                  read(fn, (void *)data, BUFF_SIZE * sizeof(short));
                } else {
                        read(fn, (void *)data, length);
                        for(i=length/sizeof(short); i<= BUFF_SIZE; i++)
                                data[i]=0;
                }
                close(fn);

                fill(31,31,30+511,30+255,0);
                drawGraph(data, 11);

                for(i=0; i<= BUFF_SIZE; i++)
                        moto[i]=data[i];
        } else {
                locate(0,28);
                puts("ファイルがオープン出来ません");
        }
}

/*
**      ファイルに保存する
*/

saveData()
{
        int     fn;

        locate(0,28);
        b_input("File Name: ",sizeof(fileName),fileName,-1);
        clrWin(1);

        fn = open(fileName, O_BINARY | O_WRONLY | O_CREAT);

        if (fn >= 0) {
                write(fn, (void *)data, BUFF_SIZE * sizeof(short));
                close(fn);
        } else {
                locate(0,28);
                puts("ファイルがオープン出来ません");
        }
}

/*
**  波形を解析する
*/
analize() {
        int     i;

        locate(0,28);
        puts("Now Calculating ...");

        for(i=1; i <= BUFF_SIZE; i++) {
                y[i].Re = moto[i+1];
                y[i].Im = 0;
        }
        fft(y,c,BUFF_SIZE,-1);

        for(i=0; i<=31; i++) {
                erasVr(i);
                s[i] = c[i+1].Re;
                drawVr(i);
        }
        for(i=32; i<=63; i++) {
                erasVr(i);
                s[i] = c[i-30].Im;
                drawVr(i);
        }

        clrWin(1);

}

/*
**  波形を合成する
*/
view() {
        int     i;

        locate(0,28);
        puts("Now Calculating ...");

        for(i=1; i <= BUFF_SIZE; i++) {
                y[i].Re = data[i];
                y[i].Im = 0;
        }

        for(i=0 ; i<=31; i++)
                c[i+1].Re = s[i];
        for(i=32; i<=63; i++)
                c[i-30].Im = s[i];

        fft(c, z, BUFF_SIZE,1);

        for(i=1; i <= BUFF_SIZE; i++) {
                data[i] = z[i].Re;
        }

        fill(31,31,30+511,30+255,0);
        drawGraph(moto, 11);
        drawGraph(data, 13);

                clrWin(1);

}
```

68881+68881=137762?

68881並列駆動への挑戦

Kuwano Masahiko 桒野 雅彦

高速に浮動小数点演算を実行してくれる68881。1台使えば高速演算，2台使えば……ということで，用意されている2つのI/Oポートを使い，数値演算プロセッサボードの並列駆動に挑戦してみます。

X68000シリーズでは，浮動小数点演算プロセッサ（MC68881：以下68881と略します）をI/Oデバイスとして接続して，細々とした68881とのやりとりはすべてプログラムで処理するようになっています。68020以降のCPUでは，浮動小数点演算命令を実行すると，CPU自体が自動的に（プログラマからは見えないところで）68881と連絡をとって演算処理を実行してくれます。そのため68881の存在自体を意識することもなく，単にCPUに浮動小数点レジスタが追加されたように見えるのですが，68000には残念ながら68881と直接接続するための信号がないため，I/Oとして接続するほかないわけです。

68881がI/Oデバイスとして接続されるため，操作こそ面倒ではありますが，アドレスを変更してやれば複数の68881を接続することも可能となります。実際，X68000でも数値演算プロセッサボード（CZ-6BP1）は基板上にあるピン設定で2種類のアドレス($E9E000～/$E9E080～）が選択できるようになっています。ただし，X68000の浮動小数点ドライバ（FLOAT3.X）が対応するのはこのうち片側($E9E000～)だけで，もう一方のアドレスは使い道もなく放置されています。

今回は，ここに目をつけてアドレスを変えた数値演算プロセッサボード2枚を拡張スロットに挿入し，2つの68881を直接操作して演算処理性能の改善が図れるか，実験をしてみることにしました。

サンプル作成

浮動小数点演算が多用される例としてすぐ思い浮かぶのは，レイトレーシングや流体力学のような重たさの極致のようなものです。今回もこのような例を扱おうかとも思ったのですが，68881を直接扱うとなると，ちょっとした演算処理でもプログラムの行数は予想以上に増加してしまい，見るのも打ち込むのもうんざりといった感じになってしまいます。ここではもう少し簡単な例として，惑星の航行モドキのシミュレーションを取り上げてみました。

3つの物体のうち，ひとつを固定しておいて(太陽とでも考えてください)，残る2つに適当な場所から適当な方向に初速度を与え，さらにそれぞれの物体どうしの間に距離の2乗に反比例するような，引力を働かせておきます。

引力や初速度などを適当に調整すると，2つの物体は太陽の周りを円や楕円を描きながら周回し始めますが，回っている物体どうしの間にも引力があるため話は簡単ではありません。前方を進んでいるものが減速され，後方にあるものが加速される結果，軌道が少しずつ変化してしまうわけです。

これをとりあえずエイヤ！　とプログラムにしてしまったのがリスト1。初期値が変な値になっているのは当初，太陽，地球，月の3つにしようと思ったときの名残です。また，ソースがCにしてはやや奇妙になっているのは，お察しのとおり，X-BASICのプログラムをBC.Xにかけて得たCのソースに手を加えて作っているためです。

そして，68881を直接アクセスするように書き換えたのがリスト2。だいぶ見苦しくなったような気がします。コーディング自体もあまりほめられたものではありませんが，速度にはあまり影響はないでしょう。計算をすべてmain( )から追い出すためと，次のステップである，68881を2個使うときのことを考えて似たような計算をしている部分を関数にまとめてみました。

そして，リスト2をさらにボード2枚化に対応させるように書き換えたのがリスト3です。68881のCIRを指すポインタが2つになるとともに，同期をとるためのwait_copro( )関数が1枚目用と2枚目用の2つに分かれます。

2枚同時に制御するとなるとwait_copro( )の呼び出し方，というか待つタイミングの考え方も少し変えなくてはなりません。1枚のときには，演算処理が終了したのを確認して次のコマンド発行にとりかかる，という方針でしたが，2枚のときには命令を与える前に前回の処理が完了しているかをチェックするようになります。つまり，片側に命令を与えたら完了を待たずに他方に命令を与え，その次に最初に命令を与えた側を再度アクセスするときに初めて同期を気にするわけです。

また68881の演算動作は応答CIRの読み出しから開始されるようですので，コマンドを与えたあとで応答CIRを読み捨てる処理をマクロで定義しています。

結果はどうなったか

さて，これらをコンパイルして走らせてみました。描画する部分（circle）はとりあえず取り払って，走りだしたときと5000回ループが終了した時点の時刻だけを表示するようにして実行します。

結果は，ごく当たり前のfloat3.Xを使用した場合が約47秒，68881が1個の場合が20秒，注目の68881が2個の場合は期待に反して1個のときと同じ20秒となってしまいました。

直接68881にアクセスすることで(エラー処理がまったくないということもありますが）2倍以上速くなったのはまず満足できる結果として，1枚と2枚で結果が同じというのは気に入りませんので，もう少し調べてみました。

その結果，wait_copro( )でCPUが待たされることはほとんどないということがわかりました。つまり，ほとんどの場合68881に命令を与えたあと，ステータスを見るとすでに処理が完了しているのです。68881を2個使って速くなるためには，68881が演算処

理の最中に次の演算命令を他方の68881に与えられるというのが前提条件です。つまり，68881に比べてCPUが十分高速でなくてはならないわけです。

ところが，X68000ではCPUのプログラム実行速度に比べて68881の演算処理速度がかなり速く，CPUが68881を待つことはほとんどないのが現状です。このため，68881を2個使っても単に交互に動作するというだけで，2つ同時には動作しておらず，処理性能はほとんど改善されないのです。

さらに今回の演算がほとんど四則演算であったということが大きな原因ではないかと考え，マニュアルをあたってみると，四則演算は50クロックから100クロック程度しかかからないことがわかりました。68000ではイミディエイト値のMOVE命令などが，ワードサイズであっても16クロック程度かかりますので，4命令程度実行している間に数値演算プロセッサ側の演算が終了しています。しかも，元祖タイプのX68000ではクロックは10MHzであるのに対し，68881は16MHzですからますますCPUの処理の遅さが目についてきます。

再挑戦してみる

かなり長いリストを作りながら残念な結果しか得られなかったわけですが，このまま引き下がるのはなんとも悔しいので，なんとか一矢報いる手を考えてみました。マニュアルを見ると，三角関数やLOGなど，超越関数関係は500クロックから600クロック程度かかるようなので，こちらを多用すれば2個使った成果が出せそうです。試しにFSINCOS命令を使って，角度を0.001から0.001刻みで変化させながらSIN値とCOS値を求める演算を50万×2回やらせてみました。極力単純にするため，内部演算だけで結果は取り出さずにやってみます。リスト4が68881を1個だけ，リスト5は68881を2個使ったプログラムです。

これは大きな効果がありました。68881が1個のときは43秒なのに対し，2個使うと26秒となり約1.7倍も高速化されました。この結果により，超越関数を多用するときは68881を複数使うことで，大きく処理速度を引き上げられることがわかります。

結論

浮動小数点の四則演算が主体の場合には，68881を2枚使ってもほとんど効果が得られないこと，より複雑な関数演算では大きな成果が得られることが実証できました。考えようによっては，四則演算が速くならない理由として，整数演算が主体の場合に68881を付けても速くはならない，というのと同じようなものだということもできるでしょう。

つまり，整数演算が主体ならCPUだけで，浮動小数点の四則演算など単純な演算なら68881が1個だけで十分であり，複雑な関数計算が主体になったときには，68881を複数使うと性能を大きく改善することができるということになります。

そして，ほとんど68881の演算ばかりというプログラムでは，CPUによる処理を100クロック程度と見積もった場合，5個程度までは同時に動かすとそれなりの効果が得られそうです。どなたかチャレンジしてみませんか？

リスト1

```
/*
 *  リスト1:float?.xを利用
 */
#define X68K     1
#define J31      0

#if     X68K
#include         <basic0.h>
#include         <basic.h>
#include         <graph.h>
#endif

static  int      cx;
static  int      cy;
static  int      mxx;
static  int      mxy;
static  double   apx;
static  double   apy;
static  double   adx;
static  double   ady;
static  double   bpx;
static  double   bpy;
static  double   bdx;
static  double   bdy;
static  double   ble2;
static  double   ale2;
static  double   xle2;
static  double   blex;
static  double   bley;
static  double   alex;
static  double   aley;
static  double   xlex;
static  double   xley;
static  double   aforce;
static  double   bforce;
static  double   xaforce;
static  double   xbforce;
static  double   afx;
static  double   bfx;
static  double   axfx;
static  double   bxfx;
static  double   afy;
static  double   bfy;
static  double   axfy;
static  double   bxfy;
static  double   cst1;
static  double   cst2;
static  int      i;


#if J31
double sqrt();
#endif
void prtime();

double calforce();


/********** program start **********/
main(b_argc,b_argv)
int      b_argc;
char     *b_argv[];
{
 screen(2,0,1,1);
 mxx=  767;mxy= 511;
 cx= 384;cy= 256;
 circle(cx,cy,5,15,'NASI','NASI','NASI');
 apx= cx    ;apy= cy-149.6 ;adx= 2.9718 ;ady= 0;
 bpx= cx-10 ;bpy= cy-139.6 ;bdx= 3.141  ;bdy= 0;
 cst1=  6.672*1.9891;cst2=  cst1*0.02; cst1=  cst1*100;
prtime();
 for(i=0 ;i<= 5000;i++){
   ble2 = callen(bpx,(double)cx,bpy,(double)cy,&blex,&bley);
   ale2 = callen(apx,(double)cx,apy,(double)cy,&alex,&aley);
   xle2 = callen(apx,bpx,apy,bpy,&xlex,&xley);
   aforce=  cst1/ale2;
   bforce=  cst1/ble2;
   xaforce=  cst2/xle2;
   xbforce=  cst2/xle2;
   afx = calforce(aforce,alex,ale2,apx,(double)cx);
   bfx=  calforce(bforce,blex,ble2,bpx,(double)cx);
   axfx= calforce(xaforce,xlex,xle2,apx,bpx);
   bxfx= calforce(xbforce,xlex,xle2,bpx,apx);
   afy=  calforce(aforce,aley,ale2,apy,(double)cy);
   bfy=  calforce(bforce,bley,ble2,bpy,(double)cy);
   axfy= calforce(xaforce,xley,xle2,apy,bpy);
   bxfy= calforce(xbforce,xley,xle2,bpy,apy);
   adx= adx+afx+axfx;ady= ady+afy+axfy;
   bdx= bdx+bfx+bxfx;bdy= bdy+bfy+bxfy;
/*   adx=adx+afx:ady=ady+afy*/
/*   bdx=bdx+bfx:bdy=bdy+bfy*/

   circle((int)(apx),(int)(apy),3,0,'NASI','NASI','NASI');
   circle((int)(bpx),(int)(bpy),3,0,'NASI','NASI','NASI');
   circle((int)(apx+adx),(int)(apy+ady),3,13,'NASI','NASI','NASI');
   circle((int)(bpx+bdx),(int)(bpy+bdy),3,15,'NASI','NASI','NASI');

   apx= apx+adx;apy= apy+ady;
   bpx= bpx+bdx;bpy= bpy+bdy;
   if( apx>mxx | apx<0 | apy>mxy | apy<0 ){break;}
   if( bpx>mxx | bpx<0 | bpy>mxy | bpy<0 ){break;}
 }
 prtime();
}

callen(px,cx,py,cy,lex,ley)
        double  px,cx,py,cy,*lex,*ley;
{
        return( (*lex = (px-cx)*(px-cx)) + (*ley = (py-cy)*(py-cy)) );
}

double calforce(force,le1,le2,p,c)
        double force,le1,le2,p,c;
{
        double  f;
        f = force * sqrt(le1/le2);
        if (p > c)
                return(-f);
        return(f);
}

#if J31
```

```
circle(x,y,l,col)
        int     x,y,l,col;
{
        crt_line(x-l,y,x+l,y,col);
        crt_line(x,y-l,x,y+l,col);
}

screen()
{
        crt_mode(0x74);
}
```

```
#endif

void prtime()
{
        int     bt;
        bt = TIMEBIN(TIMEGET());
        printf("%d %d¥n",(bt>>8)&0xff, bt & 0xff);
}


```

リスト2

```
/*
 *  リスト2：68881直接操作
 */
#include        <basic0.h>
#include        <basic.h>
#include        <graph.h>
#include        <iocslib.h>
#include        <stdio.h>

#define wait_copro(x)   while((cir->response&0xbfff)!=x)

union   DAT {
        unsigned char   cdat;
        unsigned short  sdat;
        unsigned int    idat;
        float           fdat;
        unsigned int    i2dat[2];       /* doubleの分割引き渡し用       */
        double          ddat;
} dat;

struct  CIR {
        unsigned short  response;
        unsigned short  control;
        unsigned short  save;
        unsigned short  restore;
        unsigned short  operation_word;                 /* Not used    */
        unsigned short  command;
        unsigned short  reservel;
        unsigned short  condition;
        unsigned int    operand;
        unsigned short  register_select;
        unsigned short  reserve2;
        unsigned int    instruction_address;
        unsigned int    operand_address;                /* Not used    */
};

volatile struct CIR *cir = (struct CIR *)0xe9e000;

static  int     cx;
static  int     cy;
static  int     mxx;
static  int     mxy;
static  double  dcx,dcy;
static  double  apx;
static  double  apy;
static  double  adx;
static  double  ady;
static  double  bpx;
static  double  bpy;
static  double  bdx;
static  double  bdy;
static  double  ble2;
static  double  ale2;
static  double  xle2;
static  double  blex;
static  double  bley;
static  double  alex;
static  double  aley;
static  double  xlex;
static  double  xley;
static  double  aforce;
static  double  bforce;
static  double  xaforce;
static  double  xbforce;
static  double  afx;
static  double  bfx;
static  double  axfx;
static  double  bxfx;
static  double  afy;
static  double  bfy;
static  double  axfy;
static  double  bxfy;
static  double  cst1;
static  double  cst2;
static  int     i;

double calforce();
void calf();
void caldxy();
void calpxy();
void prtime();

/********** program start **********/
main(b_argc,b_argv)
int     b_argc;
char    *b_argv[];
{
 SUPER(0);
 screen(2,0,1,1);
 mxx=  767;mxy= 511;
 cx= 384;cy= 256;
 dcx = (double)cx; dcy = (double)cy;
 circle(cx,cy,5,15,'NASI','NASI','NASI');
 apx= cx     ;apy= cy-149.6 ;adx= 2.9718 ;ady= 0;
 bpx= cx-10 ;bpy= cy-139.6 ;bdx= 3.141  ;bdy= 0;
 cst1=  6.672*1.9891;cst2=  cst1*0.02; cst1=  cst1*100;

 prtime();
 for(i=0 ;i<= 5000;i++){
   ble2 = callen(&bpx,&dcx,&bpy,&dcy,&blex,&bley);
   ale2 = callen(&apx,&dcx,&apy,&dcy,&alex,&aley);
   xle2 = callen(&apx,&bpx,&apy,&bpy,&xlex,&xley);

/*
 *  aforce=  cst1/ale2;
 *  bforce=  cst1/ble2;
 *  xaforce=  cst2/xle2;
 *  xbforce=  cst2/xle2;
 */
   calf();

   afx = calforce(&aforce,&alex,&ale2,&apx,&dcx);
   bfx=  calforce(&bforce,&blex,&ble2,&bpx,&dcx);
   axfx= calforce(&xaforce,&xlex,&xle2,&apx,&bpx);
   bxfx= calforce(&xbforce,&xlex,&xle2,&bpx,&apx);
   afy=  calforce(&aforce,&aley,&ale2,&apy,&dcy);
   bfy=  calforce(&bforce,&bley,&ble2,&bpy,&dcy);
   axfy= calforce(&xaforce,&xley,&xle2,&apy,&bpy);
   bxfy= calforce(&xbforce,&xley,&xle2,&bpy,&apy);

/*
 *  adx= adx+afx+axfx;ady= ady+afy+axfy;
 *  bdx= bdx+bfx+bxfx;bdy= bdy+bfy+bxfy;
 */
   caldxy();

   circle((int)(apx),(int)(apy),3,0,'NASI','NASI','NASI');
   circle((int)(bpx),(int)(bpy),3,0,'NASI','NASI','NASI');
   circle((int)(apx+adx),(int)(apy+ady),3,13,'NASI','NASI','NASI');
   circle((int)(bpx+bdx),(int)(bpy+bdy),3,15,'NASI','NASI','NASI');

/*
 * apx= apx+adx;apy= apy+ady;
 * bpx= bpx+bdx;bpy= bpy+bdy;
 */
   calpxy();

 }
 prtime();
}

callen(px,cx,py,cy,lex,ley)
        double  *px,*cx,*py,*cy,*lex,*ley;
{
        union DAT *dat;
        double  retdat;
/*      return( (*lex = (px-cx)*(px-cx)) + (*ley = (py-cy)*(py-cy)) );*/

        dat = px;
        cir->command = 0x5400;          /* FMOVE.D px,FP0       */
        wait_copro(0x9608);
        cir->operand = dat->i2dat[0];
        cir->operand = dat->i2dat[1];
        wait_copro(0x0802);

        dat = cx;
        cir->command = 0x5428;          /* FSUB.D  cx,FP0       */
        wait_copro(0x9608);
        cir->operand = dat->i2dat[0];
        cir->operand = dat->i2dat[1];
        wait_copro(0x0802);

        cir->command = 0x0023;          /* FMUL  FP0,FP0        */
        wait_copro(0x0802);

        dat = lex;
        cir->command = 0x7400;          /* FMOVE.D FP0,*lex     */
        wait_copro(0xb208);
        dat->i2dat[0] = cir->operand;
        dat->i2dat[1] = cir->operand;
        wait_copro(0x0802);


        dat = py;
        cir->command = 0x5480;          /* FMOVE.D py,FP1       */
        wait_copro(0x9608);
        cir->operand = dat->i2dat[0];
        cir->operand = dat->i2dat[1];
        wait_copro(0x0802);

        dat = cy;
        cir->command = 0x54a8;          /* FSUB.D  cy,FP1       */
        wait_copro(0x9608);
        cir->operand = dat->i2dat[0];
        cir->operand = dat->i2dat[1];
        wait_copro(0x0802);

        cir->command = 0x04a3;          /* FMUL  FP1,FP1        */
        wait_copro(0x0802);

        dat = ley;
        cir->command = 0x7480;          /* FMOVE.D FP1,*ley     */
        wait_copro(0xb208);
        dat->i2dat[0] = cir->operand;
        dat->i2dat[1] = cir->operand;
        wait_copro(0x0802);

        cir->command = 0x0422;          /* FADD  FP1,FP0        */
        wait_copro(0x0802);

        dat = &retdat;
        cir->command = 0x7400;          /* FMOVE.D FP0,retdat   */
        wait_copro(0xb208);
        dat->i2dat[0] = cir->operand;
        dat->i2dat[1] = cir->operand;
        wait_copro(0x0802);
```

```

	return(retdat);

}

double calforce(force,le1,le2,p,c)
	double *force,*le1,*le2,*p,*c;
{
	unsigned int	ack;
	double	f;
	union	DAT	*dat;
/*	f = force * sqrt(le1/le2); */

	dat = le1;
	cir->command = 0x5400;		/* FMOVE.D le1,FP0	*/
	wait_copro(0x9608);
	cir->operand = dat->i2dat[0];
	cir->operand = dat->i2dat[1];
	wait_copro(0x0802);

	dat = le2;
	cir->command = 0x5420;		/* FDIV.D  le2,FP0	*/
	wait_copro(0x9608);
	cir->operand = dat->i2dat[0];
	cir->operand = dat->i2dat[1];
	wait_copro(0x0802);

	cir->command = 0x0004;		/* FSQRT.D  FP0,FP0	*/
	wait_copro(0x0802);


	dat = force;
	cir->command = 0x5423;		/* FMUL.D  force,FP0	*/
	wait_copro(0x9608);
	cir->operand = dat->i2dat[0];
	cir->operand = dat->i2dat[1];
	wait_copro(0x0802);


/*
 *	if (p > c)
 *		return(-f);
 *	return(f);
 */

	dat = p;
	cir->command = 0x5480;		/* FMOVE.D p,FP1	*/
	wait_copro(0x9608);
	cir->operand = dat->i2dat[0];
	cir->operand = dat->i2dat[1];
	wait_copro(0x0802);

	dat = c;
	cir->command = 0x54a8;		/* FSUB.D  c,FP1	*/
	wait_copro(0x9608);
	cir->operand = dat->i2dat[0];
	cir->operand = dat->i2dat[1];
	wait_copro(0x0802);


	cir->condition = 0x0014;	/* LT?			*/
	do {
		ack = cir->response;
	} while((ack & 0xbffe) != 0x0800);

	if ((ack & 1) == 0) {		/* Yes			*/
		cir->command = 0x001a;	/* FNEG FP0,FP0		*/
		wait_copro(0x0802);
	}

	dat = &f;
	cir->command = 0x7400;		/* FMOVE.D FP0,f	*/
	wait_copro(0xb208);
	dat->i2dat[0] = cir->operand;
	dat->i2dat[1] = cir->operand;
	wait_copro(0x0802);

	return(f);
}


/*
 *  aforce=  cst1/ale2;
 *  bforce=  cst1/ble2;
 *  xaforce=  cst2/xle2;
 *  xbforce=  cst2/xle2;
 */
void calf()
{
	union DAT	*dat;
	dat = &cst1;
	cir->command = 0x5400;		/* FMOVE.D cst1,FP0	*/
	wait_copro(0x9608);
	cir->operand = dat->i2dat[0];
	cir->operand = dat->i2dat[1];
	wait_copro(0x0802);

	cir->command = 0x0080;		/* FMOVE  FP0,FP1	*/
	wait_copro(0x0802);

	dat = &ale2;
	cir->command = 0x5420;		/* FDIV.D  ale2,FP0	*/
	wait_copro(0x9608);
	cir->operand = dat->i2dat[0];
	cir->operand = dat->i2dat[1];
	wait_copro(0x0802);

	dat = &ble2;
	cir->command = 0x54a0;		/* FDIV.D  ble2,FP1	*/
	wait_copro(0x9608);
	cir->operand = dat->i2dat[0];
	cir->operand = dat->i2dat[1];
	wait_copro(0x0802);

	dat = &aforce;
	cir->command = 0x7400;		/* FMOVE.D FP0,aforce	*/
	wait_copro(0xb208);
	dat->i2dat[0] = cir->operand;
	dat->i2dat[1] = cir->operand;
	wait_copro(0x0802);

	dat = &bforce;
	cir->command = 0x7480;		/* FMOVE.D FP1,bforce	*/
	wait_copro(0xb208);
	dat->i2dat[0] = cir->operand;
	dat->i2dat[1] = cir->operand;
	wait_copro(0x0802);




	dat = &cst2;
	cir->command = 0x5400;		/* FMOVE.D cst2,FP0	*/
	wait_copro(0x9608);
	cir->operand = dat->i2dat[0];
	cir->operand = dat->i2dat[1];
	wait_copro(0x0802);

	cir->command = 0x0080;		/* FMOVE  FP0,FP1	*/
	wait_copro(0x0802);

	dat = &xle2;
	cir->command = 0x5420;		/* FDIV.D  xle2,FP0	*/
	wait_copro(0x9608);
	cir->operand = dat->i2dat[0];
	cir->operand = dat->i2dat[1];
	wait_copro(0x0802);

	dat = &xle2;
	cir->command = 0x54a0;		/* FDIV.D  xle2,FP1	*/
	wait_copro(0x9608);
	cir->operand = dat->i2dat[0];
	cir->operand = dat->i2dat[1];
	wait_copro(0x0802);

	dat = &xaforce;
	cir->command = 0x7400;		/* FMOVE.D FP0,xaforce	*/
	wait_copro(0xb208);
	dat->i2dat[0] = cir->operand;
	dat->i2dat[1] = cir->operand;
	wait_copro(0x0802);

	dat = &xbforce;
	cir->command = 0x7480;		/* FMOVE.D FP1,xbforce	*/
	wait_copro(0xb208);
	dat->i2dat[0] = cir->operand;
	dat->i2dat[1] = cir->operand;
	wait_copro(0x0802);
}

/*
 *  adx= adx+afx+axfx;ady= ady+afy+axfy;
 *  bdx= bdx+bfx+bxfx;bdy= bdy+bfy+bxfy;
 */
void caldxy()
{
	union DAT	*dat;

	dat = &adx;
	cir->command = 0x5400;		/* FMOVE.D adx,FP0	*/
	wait_copro(0x9608);
	cir->operand = dat->i2dat[0];
	cir->operand = dat->i2dat[1];
	wait_copro(0x0802);

	dat = &afx;
	cir->command = 0x5422;		/* FADD.D  afx,FP0	*/
	wait_copro(0x9608);
	cir->operand = dat->i2dat[0];
	cir->operand = dat->i2dat[1];
	wait_copro(0x0802);

	dat = &axfx;
	cir->command = 0x5422;		/* FADD.D  axfx,FP0	*/
	wait_copro(0x9608);
	cir->operand = dat->i2dat[0];
	cir->operand = dat->i2dat[1];
	wait_copro(0x0802);

	dat = &adx;
	cir->command = 0x7400;		/* FMOVE.D FP0,adx */
	wait_copro(0xb208);
	dat->i2dat[0] = cir->operand;
	dat->i2dat[1] = cir->operand;
	wait_copro(0x0802);


	dat = &ady;
	cir->command = 0x5400;		/* FMOVE.D ady,FP0	*/
	wait_copro(0x9608);
	cir->operand = dat->i2dat[0];
	cir->operand = dat->i2dat[1];
	wait_copro(0x0802);

	dat = &afy;
	cir->command = 0x5422;		/* FADD.D  afy,FP0	*/
	wait_copro(0x9608);
	cir->operand = dat->i2dat[0];
	cir->operand = dat->i2dat[1];
	wait_copro(0x0802);

	dat = &axfy;
	cir->command = 0x5422;		/* FADD.D  axfy,FP0	*/
	wait_copro(0x9608);
	cir->operand = dat->i2dat[0];
	cir->operand = dat->i2dat[1];
	wait_copro(0x0802);

	dat = &ady;
	cir->command = 0x7400;		/* FMOVE.D FP0,ady */
	wait_copro(0xb208);
	dat->i2dat[0] = cir->operand;
	dat->i2dat[1] = cir->operand;
	wait_copro(0x0802);



	dat = &bdx;
	cir->command = 0x5400;		/* FMOVE.D bdx,FP0	*/
	wait_copro(0x9608);
	cir->operand = dat->i2dat[0];
	cir->operand = dat->i2dat[1];
	wait_copro(0x0802);

```


▶ソフマッ○タイムス7月号を読んだ。某国民機とその互換機について，極めて正当な評価をした記事があってとても気分がよかった。さらに，X68000についての正しい認識を示した記事もあって大いに気分をよくした。真実を伝える記事は読んでいてとても気持ちがいいものだ。

藤原　利治(25)東京都

```
        dat = &bfx;
        cir->command = 0x5422;              /* FADD.D  bfx,FP0      */
        wait_copro(0x9608);
        cir->operand = dat->i2dat[0];
        cir->operand = dat->i2dat[1];
        wait_copro(0x0802);

        dat = &bxfx;
        cir->command = 0x5422;              /* FADD.D  bxfx,FP0     */
        wait_copro(0x9608);
        cir->operand = dat->i2dat[0];
        cir->operand = dat->i2dat[1];
        wait_copro(0x0802);

        dat = &bdx;
        cir->command = 0x7400;              /* FMOVE.D FP0,bdx */
        wait_copro(0xb208);
        dat->i2dat[0] = cir->operand;
        dat->i2dat[1] = cir->operand;
        wait_copro(0x0802);

        dat = &bdy;
        cir->command = 0x5400;              /* FMOVE.D bdy,FP0      */
        wait_copro(0x9608);
        cir->operand = dat->i2dat[0];
        cir->operand = dat->i2dat[1];
        wait_copro(0x0802);

        dat = &bfy;
        cir->command = 0x5422;              /* FADD.D  bfy,FP0      */
        wait_copro(0x9608);
        cir->operand = dat->i2dat[0];
        cir->operand = dat->i2dat[1];
        wait_copro(0x0802);

        dat = &bxfy;
        cir->command = 0x5422;              /* FADD.D  bxfy,FP0     */
        wait_copro(0x9608);
        cir->operand = dat->i2dat[0];
        cir->operand = dat->i2dat[1];
        wait_copro(0x0802);

        dat = &bdy;
        cir->command = 0x7400;              /* FMOVE.D FP0,bdy */
        wait_copro(0xb208);
        dat->i2dat[0] = cir->operand;
        dat->i2dat[1] = cir->operand;
        wait_copro(0x0802);
}


/*
 * apx= apx+adx;apy= apy+ady;
 * bpx= bpx+bdx;bpy= bpy+bdy;
 */
void calpxy()
{
        union   DAT     *dat;
        dat = &apx;
        cir->command = 0x5400;              /* FMOVE.D apx,FP0      */
        wait_copro(0x9608);
        cir->operand = dat->i2dat[0];
        cir->operand = dat->i2dat[1];
        wait_copro(0x0802);

        dat = &adx;
        cir->command = 0x5422;              /* FADD.D  adx,FP0      */
        wait_copro(0x9608);
        cir->operand = dat->i2dat[0];
        cir->operand = dat->i2dat[1];
        wait_copro(0x0802);

        dat = &apx;
        cir->command = 0x7400;              /* FMOVE.D FP0,apx */
        wait_copro(0xb208);
        dat->i2dat[0] = cir->operand;
        dat->i2dat[1] = cir->operand;
        wait_copro(0x0802);

        dat = &apy;
        cir->command = 0x5400;              /* FMOVE.D apy,FP0      */
        wait_copro(0x9608);
        cir->operand = dat->i2dat[0];
        cir->operand = dat->i2dat[1];
        wait_copro(0x0802);

        dat = &ady;
        cir->command = 0x5422;              /* FADD.D  ady,FP0      */
        wait_copro(0x9608);
        cir->operand = dat->i2dat[0];
        cir->operand = dat->i2dat[1];
        wait_copro(0x0802);

        dat = &apy;
        cir->command = 0x7400;              /* FMOVE.D FP0,apy */
        wait_copro(0xb208);
        dat->i2dat[0] = cir->operand;
        dat->i2dat[1] = cir->operand;
        wait_copro(0x0802);



        dat = &bpx;
        cir->command = 0x5400;              /* FMOVE.D bpx,FP0      */
        wait_copro(0x9608);
        cir->operand = dat->i2dat[0];
        cir->operand = dat->i2dat[1];
        wait_copro(0x0802);

        dat = &bdx;
        cir->command = 0x5422;              /* FADD.D  bdx,FP0      */
        wait_copro(0x9608);
        cir->operand = dat->i2dat[0];
        cir->operand = dat->i2dat[1];
        wait_copro(0x0802);

        dat = &bpx;
        cir->command = 0x7400;              /* FMOVE.D FP0,bpx */
        wait_copro(0xb208);
        dat->i2dat[0] = cir->operand;
        dat->i2dat[1] = cir->operand;
        wait_copro(0x0802);

        dat = &bpy;
        cir->command = 0x5400;              /* FMOVE.D bpy,FP0      */
        wait_copro(0x9608);
        cir->operand = dat->i2dat[0];
        cir->operand = dat->i2dat[1];
        wait_copro(0x0802);

        dat = &bdy;
        cir->command = 0x5422;              /* FADD.D  bdy,FP0      */
        wait_copro(0x9608);
        cir->operand = dat->i2dat[0];
        cir->operand = dat->i2dat[1];
        wait_copro(0x0802);

        dat = &bpy;
        cir->command = 0x7400;              /* FMOVE.D FP0,bpy */
        wait_copro(0xb208);
        dat->i2dat[0] = cir->operand;
        dat->i2dat[1] = cir->operand;
        wait_copro(0x0802);
}

void prtime()
{
        int     bt;
        bt = TIMEBIN(TIMEGET());
        printf("%d %d¥n",(bt>>8)&0xff, bt & 0xff);
}

```

リスト3

```
/*
 * リスト3：68881×2 直接操作
 */
#include        <basic0.h>
#include        <basic.h>
#include        <graph.h>
#include        <iocslib.h>
#include        <stdio.h>

#define wait_coproa(x)  while((cira->response&0xbfff)!=x)
#define wait_coprob(x)  while((cirb->response&0xbfff)!=x)
#define read_coproa()   dummy = cira->response
#define read_coprob()   dummy = cirb->response

union   DAT {
        unsigned char   cdat;
        unsigned short  sdat;
        unsigned int    idat;
        float           fdat;
        unsigned int    i2dat[2];           /* doubleの分割引き渡し用        */
        double          ddat;
};

struct  CIR {
        unsigned short  response;
        unsigned short  control;
        unsigned short  save;
        unsigned short  restore;
        unsigned short  operation_word;                     /* Not used     */
        unsigned short  command;
        unsigned short  reserve1;
        unsigned short  condition;
        unsigned int    operand;
        unsigned short  register_select;
        unsigned short  reserve2;
        unsigned int    instruction_address;
        unsigned int    operand_address;                    /* Not used     */
};

volatile struct CIR *cira = (struct CIR *)0xe9e000;
volatile struct CIR *cirb = (struct CIR *)0xe9e080;

static  int     cx;
static  int     cy;
static  int     mxx;
static  int     mxy;
static  double  dcx,dcy;
static  double  apx;
static  double  apy;
static  double  adx;
static  double  ady;
static  double  bpx;
static  double  bpy;
static  double  bdx;
static  double  bdy;
static  double  ble2;
static  double  ale2;
static  double  xle2;
static  double  blex;
static  double  bley;
static  double  alex;
static  double  aley;
static  double  xlex;
static  double  xley;
static  double  aforce;
static  double  bforce;
static  double  xaforce;
static  double  xbforce;
static  double  afx;
static  double  bfx;
static  double  axfx;
static  double  bxfx;
static  double  afy;
static  double  bfy;
static  double  axfy;
static  double  bxfy;
```

▶水谷優子さんのCD「元気」を聴いていると切なくなってしまう。ちょっとしたことに心を動かされてしまうのは，心の貧しさの現れか，それとも歳をとったからか。

坊農　誠(21)福井県

```
static  double  cst1;
static  double  cst2;
static  int     i;
static  unsigned short  dummy;

void calforce();
void calf();
void caldxy();
void calpxy();
void prtime();

/********** program start **********/
main(b_argc,b_argv)
int     b_argc;
char    *b_argv[];
{
 SUPER(0);
 screen(2,0,1,1);
 mxx=  767;mxy= 511;
 cx= 384;cy= 256;
 dcx = (double)cx; dcy = (double)cy;
 circle(cx,cy,5,15,'NASI','NASI','NASI');
 apx= cx    ;apy= cy-149.6 ;adx= 2.9718 ;ady= 0;
 bpx= cx-10 ;bpy= cy-139.6 ;bdx= 3.141  ;bdy= 0;
 cst1=  6.672*1.9891;cst2=  cst1*0.02; cst1=  cst1*100;

 prtime();
 for(i=0 ;i<= 5000;i++){
   ble2 = callen(&bpx,&dcx,&bpy,&dcy,&blex,&bley);
   ale2 = callen(&apx,&dcx,&apy,&dcy,&alex,&aley);
   xle2 = callen(&apx,&bpx,&apy,&bpy,&xlex,&xley);

/*
 *  aforce=  cst1/ale2;
 *  bforce=  cst1/ble2;
 *  xaforce=  cst2/xle2;
 *  xbforce=  cst2/xle2;
 */
   calf();

   calforce(&aforce,&alex,&ale2,&apx,&dcx,&afx,
            &bforce,&blex,&ble2,&bpx,&dcx,&bfx);

   calforce(&xaforce,&xlex,&xle2,&apx,&bpx,&axfx,
            &xbforce,&xlex,&xle2,&bpx,&apx,&bxfx);
   calforce(&aforce,&aley,&ale2,&apy,&dcy,&afy,
            &bforce,&bley,&ble2,&bpy,&dcy,&bfy);
   calforce(&xaforce,&xley,&xle2,&apy,&bpy,&axfy,
            &xbforce,&xley,&xle2,&bpy,&apy,&bxfy);

/*
 *  adx= adx+afx+axfx;ady= ady+afy+axfy;
 *  bdx= bdx+bfx+bxfx;bdy= bdy+bfy+bxfy;
 */
   caldxy();

   circle((int)(apx),(int)(apy),3,0,'NASI','NASI','NASI');
   circle((int)(bpx),(int)(bpy),3,0,'NASI','NASI','NASI');
   circle((int)(apx+adx),(int)(apy+ady),3,13,'NASI','NASI','NASI');
   circle((int)(bpx+bdx),(int)(bpy+bdy),3,15,'NASI','NASI','NASI');

/*
 * apx= apx+adx;apy= apy+ady;
 * bpx= bpx+bdx;bpy= bpy+bdy;
 */
   calpxy();

 }
 prtime();
}

callen(px,cx,py,cy,lex,ley)
        double  *px,*cx,*py,*cy,*lex,*ley;
{
        union DAT *dat;
        double  retdat;
/*      return( (*lex = (px-cx)*(px-cx)) + (*ley = (py-cy)*(py-cy)) ); */

        cira->command = 0x5400;         /* FaMOVE.D px,FP0      */
        cirb->command = 0x5480;         /* FbMOVE.D py,FP1      */

        dat = px;
        wait_copora(0x9608);
        cira->operand = dat->i2dat[0];
        cira->operand = dat->i2dat[1];
        read_copora();

        dat = py;
        wait_coprob(0x9608);
        cirb->operand = dat->i2dat[0];
        cirb->operand = dat->i2dat[1];
        read_coprob();


        wait_copora(0x0802);
        cira->command = 0x5428;         /* FaSUB.D  cx,FP0      */

        wait_coprob(0x0802);
        cirb->command = 0x54a8;         /* FbSUB.D  cy,FP1      */

        dat = cx;
        wait_copora(0x9608);
        cira->operand = dat->i2dat[0];
        cira->operand = dat->i2dat[1];
        read_copora();

        dat = cy;
        wait_coprob(0x9608);
        cirb->operand = dat->i2dat[0];
        cirb->operand = dat->i2dat[1];
        read_coprob();


        wait_copora(0x0802);
        cira->command = 0x0023;         /* FaMUL  FP0,FP0       */
        read_copora();

        wait_coprob(0x0802);
        cirb->command = 0x04a3;         /* FbMUL  FP1,FP1       */
        read_coprob();

        wait_copora(0x0802);
        cira->command = 0x7400;         /* FaMOVE.D FP0,*lex    */
        read_copora();

        wait_coprob(0x0802);
        cirb->command = 0x7480;         /* FbMOVE.D FP1,*ley    */
        read_coprob();

        wait_copora(0xb208);
        dat = lex;
        dat->i2dat[0] = cira->operand;
        dat->i2dat[1] = cira->operand;

        wait_coprob(0xb208);
        dat = ley;
        dat->i2dat[0] = cirb->operand;
        dat->i2dat[1] = cirb->operand;

        wait_copora(0x0802);
        cira->command = 0x5422;         /* FaADD.D  *ley,FP0    */
        wait_copora(0x9608);
        cira->operand = dat->i2dat[0];
        cira->operand = dat->i2dat[1];
        wait_coprob(0x0802);
        wait_copora(0x0802);

        cira->command = 0x7400;         /* FaMOVE.D FP0,retdat  */
        wait_copora(0xb208);
        dat = &retdat;
        dat->i2dat[0] = cira->operand;
        dat->i2dat[1] = cira->operand;
        wait_copora(0x0802);

        return(retdat);

}

void calforce(forcea,le1a,le2a,pa,ca,fa,forceb,le1b,le2b,pb,cb,fb)
        double *forcea,*le1a,*le2a,*pa,*ca,*forceb,*le1b,*le2b,*pb,*cb;
        double *fa,*fb;
{
        unsigned int    ack;
        union   DAT     *dat;
/*      f = force * sqrt(le1/le2); */

        cira->command = 0x5400;         /* FaMOVE.D le1a,FP0    */
        cirb->command = 0x5400;         /* FbMOVE.D le1b,FP0    */

        dat = le1a;
        wait_copora(0x9608);
        cira->operand = dat->i2dat[0];
        cira->operand = dat->i2dat[1];
        read_copora();

        dat = le1b;
        wait_coprob(0x9608);
        cirb->operand = dat->i2dat[0];
        cirb->operand = dat->i2dat[1];
        read_coprob();

        wait_copora(0x0802);
        cira->command = 0x5420;         /* FaDIV.D  le2a,FP0    */

        wait_coprob(0x0802);
        cirb->command = 0x5420;         /* FbDIV.D  le2b,FP0    */

        wait_copora(0x9608);
        dat = le2a;
        cira->operand = dat->i2dat[0];
        cira->operand = dat->i2dat[1];
        read_copora();


        dat = le2b;
        wait_coprob(0x9608);
        cirb->operand = dat->i2dat[0];
        cirb->operand = dat->i2dat[1];
        read_coprob();


        wait_copora(0x0802);
        cira->command = 0x0004;         /* FaSQRT.D  FP0,FP0    */
        read_copora();

        wait_coprob(0x0802);
        cirb->command = 0x0004;         /* FbSQRT.D  FP0,FP0    */
        read_coprob();


        wait_copora(0x0802);
        cira->command = 0x5423;         /* FaMUL.D  forcea,FP0  */

        wait_coprob(0x0802);
        cirb->command = 0x5423;         /* FbMUL.D  forceb,FP0  */


        dat = forcea;
        wait_copora(0x9608);
        cira->operand = dat->i2dat[0];
        cira->operand = dat->i2dat[1];
        read_copora();

        dat = forceb;
        wait_coprob(0x9608);
        cirb->operand = dat->i2dat[0];
        cirb->operand = dat->i2dat[1];
        read_coprob();


/*
 *      if (p > c)
 *              f = -f;
 */

        wait_copora(0x0802);
        cira->command = 0x5480;         /* FaMOVE.D pa,FP1      */

        wait_coprob(0x0802);
        cirb->command = 0x5480;         /* FbMOVE.D pb,FP1      */


        dat = pa;
        wait_copora(0x9608);
```


▶「MATIER」が面白そう。後編の記事を楽しみにしています。19,800円くらいなら迷わず買っちゃうのになあ。

林 裕司(16)福島県

```
	cira->operand = dat->i2dat[0];
	cira->operand = dat->i2dat[1];
	read_coproa();

	dat = pb;
	wait_coprob(0x9608);
	cirb->operand = dat->i2dat[0];
	cirb->operand = dat->i2dat[1];
	read_coprob();

	wait_coproa(0x0802);
	cira->command = 0x54a8;			/* FaSUB.D  ca,FP1	*/

	wait_coprob(0x0802);
	cirb->command = 0x54a8;			/* FbSUB.D  cb,FP1	*/


	dat = ca;
	wait_coproa(0x9608);
	cira->operand = dat->i2dat[0];
	cira->operand = dat->i2dat[1];
	read_coproa();

	dat = cb;
	wait_coprob(0x9608);
	cirb->operand = dat->i2dat[0];
	cirb->operand = dat->i2dat[1];
	read_coprob();

	wait_coproa(0x0802);
	cira->condition = 0x0014;		/* LT?			*/
	do {
		ack = cira->response;
	} while((ack & 0xbffe) != 0x0800);


	if ((ack & 1) == 0) {			/* Yes			*/
		cira->command = 0x001a;	/* FaNEG FP0,FP0	*/
		read_coproa();
	}

	wait_coprob(0x0802);
	cirb->condition = 0x0014;		/* LT?			*/
	do {
		ack = cirb->response;
	} while((ack & 0xbffe) != 0x0800);


	if ((ack & 1) == 0) {			/* Yes			*/
		cirb->command = 0x001a;	/* FbNEG FP0,FP0	*/
		read_coprob();
	}


	wait_coproa(0x0802);
	cira->command = 0x7400;			/* FaMOVE.D FP0,fa	*/
	read_coproa();

	wait_coprob(0x0802);
	cirb->command = 0x7400;			/* FbMOVE.D FP0,fb	*/
	read_coprob();

	dat = fa;
	wait_coproa(0xb208);
	dat->i2dat[0] = cira->operand;
	dat->i2dat[1] = cira->operand;


	dat = fb;
	wait_coprob(0xb208);
	dat->i2dat[0] = cirb->operand;
	dat->i2dat[1] = cirb->operand;

	wait_coproa(0x0802);
	wait_coprob(0x0802);
}


/*
 *  aforce=  cst1/ale2;
 *  bforce=  cst1/ble2;
 *  xaforce=  cst2/xle2;
 *  xbforce=  cst2/xle2;
 */
void calf()
{
	union DAT	*dat;

	cira->command = 0x5400;			/* FaMOVE.D cst1,FP0	*/
	cirb->command = 0x5400;			/* FbMOVE.D cst2,FP0	*/

	dat = &cst1;
	wait_coproa(0x9608);
	cira->operand = dat->i2dat[0];
	cira->operand = dat->i2dat[1];
	read_coproa();

	dat = &cst2;
	wait_coprob(0x9608);
	cirb->operand = dat->i2dat[0];
	cirb->operand = dat->i2dat[1];
	read_coprob();

	wait_coproa(0x0802);
	cira->command = 0x0080;			/* FaMOVE   FP0,FP1	*/
	read_coproa();

	wait_coprob(0x0802);
	cirb->command = 0x0080;			/* FbMOVE   FP0,FP1	*/
	read_coprob();

	wait_coproa(0x0802);
	cira->command = 0x5420;			/* FaDIV.D  ale2,FP0	*/

	wait_coprob(0x0802);
	cirb->command = 0x5420;			/* FbDIV.D  xle2,FP0	*/

	dat = &ale2;
	wait_coproa(0x9608);
	cira->operand = dat->i2dat[0];
	cira->operand = dat->i2dat[1];
	read_coproa();

	dat = &xle2;
	wait_coprob(0x9608);
	cirb->operand = dat->i2dat[0];
	cirb->operand = dat->i2dat[1];
	read_coprob();

	wait_coproa(0x0802);
	cira->command = 0x54a0;			/* FaDIV.D  ble2,FP1	*/

	wait_coprob(0x0802);
	cirb->command = 0x54a0;			/* FbDIV.D  xle2,FP1	*/

	dat = &ble2;
	wait_coproa(0x9608);
	cira->operand = dat->i2dat[0];
	cira->operand = dat->i2dat[1];
	read_coproa();

	dat = &xle2;
	wait_coprob(0x9608);
	cirb->operand = dat->i2dat[0];
	cirb->operand = dat->i2dat[1];
	read_coprob();

	wait_coproa(0x0802);
	cira->command = 0x7400;			/* FaMOVE.D FP0,aforce	*/
	read_coproa();

	wait_coprob(0x0802);
	cirb->command = 0x7400;			/* FbMOVE.D FP0,xaforce */
	read_coprob();

	dat = &aforce;
	wait_coproa(0xb208);
	dat->i2dat[0] = cira->operand;
	dat->i2dat[1] = cira->operand;

	dat = &xaforce;
	wait_coprob(0xb208);
	dat->i2dat[0] = cirb->operand;
	dat->i2dat[1] = cirb->operand;

	wait_coproa(0x0802);
	cira->command = 0x7480;			/* FaMOVE.D FP1,bforce	*/
	read_coproa();

	wait_coprob(0x0802);
	cirb->command = 0x7480;			/* FbMOVE.D FP1,xbforce */
	read_coprob();

	dat = &bforce;
	wait_coproa(0xb208);
	dat->i2dat[0] = cira->operand;
	dat->i2dat[1] = cira->operand;

	dat = &xbforce;
	wait_coprob(0xb208);
	dat->i2dat[0] = cirb->operand;
	dat->i2dat[1] = cirb->operand;

	wait_coproa(0x0802);
	wait_coprob(0x0802);
}

/*
 *  adx= adx+afx+axfx;ady= ady+afy+axfy;
 *  bdx= bdx+bfx+bxfx;bdy= bdy+bfy+bxfy;
 */
void caldxy()
{
	union DAT	*dat;

	cira->command = 0x5400;			/* FaMOVE.D adx,FP0	*/

	cirb->command = 0x5400;			/* FbMOVE.D bdx,FP0	*/

	dat = &adx;
	wait_coproa(0x9608);
	cira->operand = dat->i2dat[0];
	cira->operand = dat->i2dat[1];
	read_coproa();

	dat = &bdx;
	wait_coprob(0x9608);
	cirb->operand = dat->i2dat[0];
	cirb->operand = dat->i2dat[1];
	read_coprob();


	wait_coproa(0x0802);
	cira->command = 0x5422;			/* FaADD.D  afx,FP0	*/

	wait_coprob(0x0802);
	cirb->command = 0x5422;			/* FbADD.D  bfx,FP0	*/

	dat = &afx;
	wait_coproa(0x9608);
	cira->operand = dat->i2dat[0];
	cira->operand = dat->i2dat[1];
	read_coproa();

	dat = &bfx;
	wait_coprob(0x9608);
	cirb->operand = dat->i2dat[0];
	cirb->operand = dat->i2dat[1];
	read_coprob();

	wait_coproa(0x0802);
	cira->command = 0x5422;			/* FaADD.D  axfx,FP0	*/

	wait_coprob(0x0802);
	cirb->command = 0x5422;			/* FbADD.D  bxfx,FP0	*/

	dat = &axfx;
	wait_coproa(0x9608);
	cira->operand = dat->i2dat[0];
	cira->operand = dat->i2dat[1];
	read_coproa();

```

▶初めてパソコンに触れた日から9年ほどたちました。当時使っていたMZ-2000が眠ってから，まともにプログラムを組んでいません。現在，X68000にはかわいそうな思いをさせています。アッコちょんを見習ってプログラムを組もう！　受験勉強の合間にできるでしょう。できたら投稿します。　古川　とおる(17)兵庫県

```c
        dat = &bxfx;
        wait_coprob(0x9608);
        cirb->operand = dat->i2dat[0];
        cirb->operand = dat->i2dat[1];
        read_coprob();


        wait_coproa(0x0802);
        cira->command = 0x7400;          /* FaMOVE.D FP0,adx */
        read_coproa();

        wait_coprob(0x0802);
        cirb->command = 0x7400;          /* FbMOVE.D FP0,bdx */
        read_coprob();


        dat = &adx;
        wait_coproa(0xb208);
        dat->i2dat[0] = cira->operand;
        dat->i2dat[1] = cira->operand;

        dat = &bdx;
        wait_coprob(0xb208);
        dat->i2dat[0] = cirb->operand;
        dat->i2dat[1] = cirb->operand;

        wait_coproa(0x0802);
        cira->command = 0x5400;          /* FaMOVE.D ady,FP0     */

        wait_coprob(0x0802);
        cirb->command = 0x5400;          /* FbMOVE.D bdy,FP0     */

        dat = &ady;
        wait_coproa(0x9608);
        cira->operand = dat->i2dat[0];
        cira->operand = dat->i2dat[1];
        read_coproa();

        dat = &bdy;
        wait_coprob(0x9608);
        cirb->operand = dat->i2dat[0];
        cirb->operand = dat->i2dat[1];
        read_coprob();

        wait_coproa(0x0802);
        cira->command = 0x5422;          /* FaADD.D  afy,FP0     */

        wait_coprob(0x0802);
        cirb->command = 0x5422;          /* FbADD.D  bfy,FP0     */

        dat = &afy;
        wait_coproa(0x9608);
        cira->operand = dat->i2dat[0];
        cira->operand = dat->i2dat[1];
        read_coproa();

        dat = &bfy;
        wait_coprob(0x9608);
        cirb->operand = dat->i2dat[0];
        cirb->operand = dat->i2dat[1];
        read_coprob();

        wait_coproa(0x0802);
        cira->command = 0x5422;          /* FaADD.D  axfy,FP0    */

        wait_coprob(0x0802);
        cirb->command = 0x5422;          /* FbADD.D  bxfy,FP0    */

        dat = &axfy;
        wait_coproa(0x9608);
        cira->operand = dat->i2dat[0];
        cira->operand = dat->i2dat[1];
        read_coproa();

        dat = &bxfy;
        wait_coprob(0x9608);
        cirb->operand = dat->i2dat[0];
        cirb->operand = dat->i2dat[1];
        read_coprob();

        wait_coproa(0x0802);
        cira->command = 0x7400;          /* FaMOVE.D FP0,ady */
        read_coproa();

        wait_coprob(0x0802);
        cirb->command = 0x7400;          /* FbMOVE.D FP0,bdy */
        read_coprob();

        dat = &ady;
        wait_coproa(0xb208);
        dat->i2dat[0] = cira->operand;
        dat->i2dat[1] = cira->operand;

        dat = &bdy;
        wait_coprob(0xb208);
        dat->i2dat[0] = cirb->operand;
        dat->i2dat[1] = cirb->operand;

        wait_coproa(0x0802);
        wait_coprob(0x0802);
}


/*
 * apx= apx+adx;apy= apy+ady;
 * bpx= bpx+bdx;bpy= bpy+bdy;
 */
void calpxy()
{
        union   DAT     *dat;

        cira->command = 0x5400;          /* FaMOVE.D apx,FP0     */

        cirb->command = 0x5400;          /* FbMOVE.D bpx,FP0     */

        dat = &apx;
        wait_coproa(0x9608);
        cira->operand = dat->i2dat[0];
        cira->operand = dat->i2dat[1];
        read_coproa();

        dat = &bpx;
        wait_coprob(0x9608);
        cirb->operand = dat->i2dat[0];
        cirb->operand = dat->i2dat[1];
        read_coprob();


        wait_coproa(0x0802);
        cira->command = 0x5422;          /* FaADD.D  adx,FP0     */

        wait_coprob(0x0802);
        cirb->command = 0x5422;          /* FbADD.D  bdx,FP0     */

        dat = &adx;
        wait_coproa(0x9608);
        cira->operand = dat->i2dat[0];
        cira->operand = dat->i2dat[1];
        read_coproa();


        dat = &bdx;
        wait_coprob(0x9608);
        cirb->operand = dat->i2dat[0];
        cirb->operand = dat->i2dat[1];
        read_coprob();


        wait_coproa(0x0802);
        cira->command = 0x7400;          /* FaMOVE.D FP0,apx */
        read_coproa();

        wait_coprob(0x0802);
        cirb->command = 0x7400;          /* FbMOVE.D FP0,bpx */
        read_coprob();


        dat = &apx;
        wait_coproa(0xb208);
        dat->i2dat[0] = cira->operand;
        dat->i2dat[1] = cira->operand;

        dat = &bpx;
        wait_coprob(0xb208);
        dat->i2dat[0] = cirb->operand;
        dat->i2dat[1] = cirb->operand;


        wait_coproa(0x0802);
        cira->command = 0x5400;          /* FaMOVE.D apy,FP0     */

        wait_coprob(0x0802);
        cirb->command = 0x5400;          /* FbMOVE.D bpy,FP0     */

        dat = &apy;
        wait_coproa(0x9608);
        cira->operand = dat->i2dat[0];
        cira->operand = dat->i2dat[1];
        read_coproa();

        dat = &bpy;
        wait_coprob(0x9608);
        cirb->operand = dat->i2dat[0];
        cirb->operand = dat->i2dat[1];
        read_coprob();

        wait_coproa(0x0802);
        cira->command = 0x5422;          /* FaADD.D  ady,FP0     */

        wait_coprob(0x0802);
        cirb->command = 0x5422;          /* FbADD.D  bdy,FP0     */

        dat = &ady;
        wait_coproa(0x9608);
        cira->operand = dat->i2dat[0];
        cira->operand = dat->i2dat[1];
        read_coproa();

        dat = &bdy;
        wait_coprob(0x9608);
        cirb->operand = dat->i2dat[0];
        cirb->operand = dat->i2dat[1];
        read_coprob();

        wait_coproa(0x0802);
        cira->command = 0x7400;          /* FaMOVE.D FP0,apy */
        read_coproa();

        wait_coprob(0x0802);
        cirb->command = 0x7400;          /* FbMOVE.D FP0,bpy */
        read_coprob();

        dat = &apy;
        wait_coproa(0xb208);
        dat->i2dat[0] = cira->operand;
        dat->i2dat[1] = cira->operand;

        dat = &bpy;
        wait_coprob(0xb208);
        dat->i2dat[0] = cirb->operand;
        dat->i2dat[1] = cirb->operand;

        wait_coproa(0x0802);
        wait_coprob(0x0802);


}


void prtime()
{
        int     bt;
        bt = TIMEBIN(TIMEGET());
        printf("%d %d¥n",(bt>>8)&0xff, bt & 0xff);
}
```


▶SX-WINDOWのアプリケーション作成ユーザーのことを「SXer」と呼ぶそうです。その例にならうとOh!Xの読者は，さしずめ「Oh!Xer」でしょうか。でもって，読み方は「オー！　エクサー」，縮めて「オエッ！　臭～」。1987年8月号180ページ以来進歩のない私でした。
野田　敏之(21)神奈川県

リスト4

```c
/*
 * リスト4
 * sin(0.000),cos(0.000)~sin(1000.000),cos(1000.000)の計算
 */

#include        <iocslib.h>
#include        <stdio.h>

#define wait_copro(x)   while((cir->response&0xbfff)!=x);

union   DAT {
        unsigned char   cdat;
        unsigned short  sdat;
        unsigned int    idat;
        float           fdat;
        double          ddat;
} dat;

struct  CIR {
        unsigned short  response;
        unsigned short  control;
        unsigned short  save;
        unsigned short  restore;
        unsigned short  operation_word;                 /* Not used     */
        unsigned short  command;
        unsigned short  reserve1;
        unsigned short  condition;
        unsigned int    operand;
        unsigned short  register_select;
        unsigned short  reserve2;
        unsigned int    instruction_address;
        unsigned int    operand_address;                /* Not used     */
};

volatile struct CIR *cir = (struct CIR *)0xe9e000;

void main();
void prtime();

void main()
{
        unsigned int    i;
        SUPER(0);
        cir->command = 0x4400;                  /* FMOVE.S #0.0,FP0     */
        dat.fdat = 0.0;
        wait_copro(0x9504);
        cir->operand = dat.idat;
        wait_copro(0x0802);

        cir->command = 0x4480;                  /* FMOVE.S #0.001,FP1   */
        dat.fdat = 0.001;
        wait_copro(0x9504);
        cir->operand = dat.idat;
        wait_copro(0x0802);

        prtime();
        for (i=0; i<500000; i++) {
                cir->command = 0x0133;          /* FSINCOS.S  FP0,FP2,FP3 */
                wait_copro(0x0802);

                cir->command = 0x0422;          /* FADD FP1,FP0         */
                wait_copro(0x0802);

                cir->command = 0x0133;          /* FSINCOS.S  FP0,FP2,FP3 */
                wait_copro(0x0802);

                cir->command = 0x0422;          /* FADD FP1,FP0         */
                wait_copro(0x0802);

        }
        prtime();
        cir->command = 0x6400;                  /* FMOVE.S FP0,xxx      */
        wait_copro(0xb104);
        dat.idat = cir->operand;
        wait_copro(0x0802);
        printf("ANG = %f¥n",dat.fdat);

        cir->command = 0x6500;                  /* FMOVE.S FP2,xxx      */
        wait_copro(0xb104);
        dat.idat = cir->operand;
        wait_copro(0x0802);
        printf("SIN = %f¥n",dat.fdat);

        cir->command = 0x6580;                  /* FMOVE.S FP3,xxx      */
        wait_copro(0xb104);
        dat.idat = cir->operand;
        wait_copro(0x0802);
        printf("COS = %f¥n",dat.fdat);
}
/*
void wait_copro(response)
        unsigned short  response;
{
        while((cir->response & 0xbfff) != response)
                ;
}
*/
void prtime()
{
        int     bt;
        bt = TIMEBIN(TIMEGET());
        printf("%d %d¥n",(bt>>8)&0xff, bt & 0xff);
}
```

リスト5

```c
/*
 * リスト5
 * sin(0.000),cos(0.000)~sin(500.000),cos(500.000)の計算*2
 */

#include        <iocslib.h>
#include        <stdio.h>

#define wait_coproa(x)  while((cira->response&0xbfff)!=x);
#define wait_coprob(x)  while((cirb->response&0xbfff)!=x);
#define read_coproa()   s = cira->response;
#define read_coprob()   s = cirb->response;

union   DAT {
        unsigned char   cdat;
        unsigned short  sdat;
        unsigned int    idat;
        float           fdat;
        double          ddat;
} dat;

struct  CIR {
        unsigned short  response;
        unsigned short  control;
        unsigned short  save;
        unsigned short  restore;
        unsigned short  operation_word;                 /* Not used     */
        unsigned short  command;
        unsigned short  reserve1;
        unsigned short  condition;
        unsigned int    operand;
        unsigned short  register_select;
        unsigned short  reserve2;
        unsigned int    instruction_address;
        unsigned int    operand_address;                /* Not used     */
};

volatile struct CIR *cira = (struct CIR *)0xe9e000;
volatile struct CIR *cirb = (struct CIR *)0xe9e080;

void main();
/*
void wait_coproa();
void wait_coprob();
*/
void prtime();

/*
  #define       wait_coproa(x)  while((cira->res&0xbfff)!=x);
  #define       wait_coprob(x)  while((cirb->res&0xbfff)!=x);
*/

void main()
{
        unsigned int    i;
        unsigned short  s;
        SUPER(0);
        cira->command = 0x4400;                 /* FMOVE.S #0.0,FP0     */
        dat.fdat = 0.0;
        wait_coproa(0x9504);
        cira->operand = dat.idat;
        wait_coproa(0x0802);

        cira->command = 0x4480;                 /* FMOVE.S #0.001,FP1   */
        dat.fdat = 0.001;
        wait_coproa(0x9504);
        cira->operand = dat.idat;
        wait_coproa(0x0802);

        cirb->command = 0x4400;                 /* FMOVE.S #0.0,FP0     */
        dat.fdat = 0.0;
        wait_coprob(0x9504);
        cirb->operand = dat.idat;
        wait_coprob(0x0802);

        cirb->command = 0x4480;                 /* FMOVE.S #0.001,FP1   */
        dat.fdat = 0.001;
        wait_coprob(0x9504);
        cirb->operand = dat.idat;
        wait_coprob(0x0802);

        prtime();
        for (i=0; i<500000; i++) {
                wait_coproa(0x0802);
                cira->command = 0x0133;         /* FSINCOS.S  FP0,FP2,FP3 */
                read_coproa();

                wait_coprob(0x0802);
                cirb->command = 0x0133;         /* FSINCOS.S  FP0,FP2,FP3 */
                read_coprob();

                wait_coproa(0x0802);
                cira->command = 0x0422;         /* FADD FP1,FP0         */
                read_coprob();

                wait_coprob(0x0802);
                cirb->command = 0x0422;         /* FADD FP1,FP0         */
                read_coprob();

        }
        prtime();


        wait_coproa(0x0802);
        wait_coprob(0x0802);
        cira->command = 0x6400;                 /* FMOVE.S FP0,xxx      */
        wait_coproa(0xb104);
        dat.idat = cira->operand;
        wait_coproa(0x0802);
        printf("ANG = %f¥n",dat.fdat);

        cira->command = 0x6500;                 /* FMOVE.S FP2,xxx      */
        wait_coproa(0xb104);
        dat.idat = cira->operand;
        wait_coproa(0x0802);
        printf("SIN = %f¥n",dat.fdat);

        cira->command = 0x6580;                 /* FMOVE.S FP3,xxx      */
        wait_coproa(0xb104);
        dat.idat = cira->operand;
        wait_coproa(0x0802);
        printf("COS = %f¥n",dat.fdat);
}
/*
void wait_coproa(res)
        unsigned short  res;
{
        while ((cira->response & 0xbfff) != res)
                ;
}

void wait_coprob(res)
        unsigned short  res;
{
        while ((cirb->response & 0xbfff) != res)
                ;
}
*/
void prtime()
{
        int     bt;
        bt = TIMEBIN(TIMEGET());
        printf("%d %d¥n",(bt>>8)&0xff, bt & 0xff);
}
```

▶ そろそろX68000を封印しなきゃ，と思っていたら８月号のプレゼント当選者欄に名前が載っていた。これで浪人になる確率が上昇するのでしょうね（実は嬉しい）。

大谷　洋徳(18)広島県

バックナンバー案内

BACK ISSUES

ここには1991年9月号から1992年8月号までをご紹介しました。現在1991年1，5，9，11，12，1992年1～8月号の在庫がございます。バックナンバーおよび定期購読の申し込み方法については，137ページを参照してください。

1991

9月号
特集　Brush up your MAGIC.

連載　マシン語プログラミング/DōGA/Z80's Bar/ショートプロ
響子 in CGわ～るど/ハード工作/シミュレーション入門
吾輩はX68000である/大人のためのX68000/C言語

●X1用ゲーム Manual Runner
●ANOTHER CG WORLD
LIVE in '91　One/WHITE MANE
THE SOFTOUCH　イース/生中継68/アークス・オデッセイ他
全機種共通システム　SLANG用NEWファイル入出力ライブラリ

10月号（品切れ）
特集　マシン語との邂逅

連載　響子 in CGわ～るど/マシン語プログラミング/ショートプロ
ハード工作/Z80's Bar/よいこのSX-WINDOW/ANOTHER CG WORLD
吾輩はX68000である/ようこそC言語/大人のためのX68000

●新連載　Computer Music入門
●NEW Print Shop PRO-68K Ver 2.0
LIVE in '91　うれしい！ たのしい！ 大好き/SPANISH BLUE
THE SOFTOUCH　ボナンザブラザーズ/ロードス島戦記/ジーザスII他
全機種共通システム　Small-C活用講座（初級編）

11月号
特集　空間彷徨型ゲーム大分析

連載　響子 in CGわ～るど/大人のためのX68000/ANOTHER CG WORLD
DōGA/ショートプロ/Computer Music入門/吾輩はX68000である
ようこそC言語/マシン語プログラミング/Z80's Bar/ハード工作

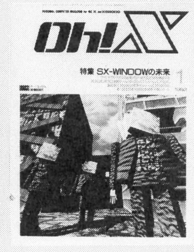
●X68000用カードゲーム　ギャップ
●新製品紹介　F-Card GT
LIVE in '91　オーダイン
THE SOFTOUCH　キャメルトライ/アクアレス/フューチャーウォーズ他
全機種共通システム　Small-C活用講座（応用編）/MORTAL

12月号
特集　音・そして音楽とコンピュータ
別冊付録　X68000 THE GAME SOFTWARE BEST SELECTION

連載　響子 in CGわ～るど/マシン語プログラミング/ショートプロ
ハード工作/Z80's Bar/ようこそC言語/ANOTHER CG WORLD
吾輩はX68000である/Computer Music入門/大人のためのX68000

●エレクトロニクスショウ＆データショウ
LIVE in '91　OH YEAH!/サイレント・イヴ/ジングルベル
THE SOFTOUCH　フェアリーランドストーリー/プロサッカー68他
全機種共通システム　Small-C用　SLANGコンパチ関数

1992



1月号
特集　SX-WINDOWの未来

連載　響子 in CGわ～るど/DōGA・CGA/大人のためのX68000
ハード工作/Z80's Bar/ショートプロ/吾輩はX68000である
ANOTHER CG WORLD/Computer Music入門/カードゲーム

●MAGIC用ゲーム　3DMAZE
●CM-300/500&LA音源の活用法
LIVE in '92　DRAGON SABER/すき/THE ENTRETAINER
THE SOFTOUCH　出たな!! ツインビー/ブリッツクリーク/飛翔鮫他
全機種共通システム　パズルゲームLINER

2月号
特集　2Dグラフィックの拡張

連載　響子 in CGわ～るど/大人のためのX68000/マシン語プログラミング
ハード工作/ショートプロ/ANOTHER CG WORLD/Z80's Bar
吾輩はX68000である/Computer Music入門/カードゲーム

●TREND ANALYSIS
●MIRAGE MODEL STUFF/Press Conductor PRO-68K
LIVE in '92　ストリートファイターII/Tide Over
THE SOFTOUCH　ジェノサイド2/アルシャーク/コード・ゼロ他
全機種共通システム　シミュレーションゲームPOLANYI

3月号
特集　SCSIの活用

連載　響子 in CGわ～るど/DōGA・CGA/大人のためのX68000/Z80's Bar
ショートプロ/吾輩はX68000である/マシン語プログラミング
ハード工作/ANOTHER CG WORLD/Computer Music入門/カードゲーム

●Z-MUSIC支援ツール　ZPDCON.X
●Z's-EX用拡張コマンド　MASK_reverse
LIVE in '92　ギャラクシーフォース/君が代
THE SOFTOUCH　グラディウスII/レミングス/大戦略/III'90/伊忍者
全機種共通システム　カードゲームKLONDIKE

4月号
特集　成熟するゲームと日本の文化

連載　よい子のSX-WINDOW/大人のためのX68000/Z80's Bar
響子 in CGわ～るど/ショートプロ/吾輩はX68000である
ハード工作/ANOTHER CG WORLD/Computer Music入門

●発表　1991年度GAME OF THE YEAR
●バーコードバトラー
LIVE in '92　あじさいのうた/ショパン練習曲作品25-2へ短調/It's MAGIC
THE SOFTOUCH　ファーストクィーンII/マスターオブモンスターズII他
全機種共通システム　実践Small-C(1)オプティマイザ080

5月号
特集　明日のための環境づくり
第7回　言わせてくれなくちゃだワ

連載　響子 in CGわ～るど/大人のためのX68000/Z80's Bar
ハード工作/ショートプロ/マシン語プログラミング
Computer Music入門/吾輩はX68000である

●製品紹介　MIDI音源 03R/W/MIC68K
LIVE in '92　フレンズ/Danger Line
THE SOFTOUCH　エイリアンシンドローム/苦胃頭捕物帳他
全機種共通システム　実践Small-C(2)COMMAND.OBJ

6月号
特別企画　Oh!MZ,Oh!X10年間の歩み
特別付録　創刊10周年記念PRO-68K(5"2HD)

連載　響子 in CGわ～るど/大人のためのX68000/マシン語プログラミング
ハード工作/ショートプロ/ANOTHER CG WORLD/Z80's Bar
吾輩はX68000である/Computer Music入門

●新製品紹介　Z'sSTAFF PRO-68K ver.3.0
LIVE in '92　Shake the Street/Ancient relics
THE SOFTOUCH　スピンディジーII/ロイヤルブラッド/ライフ&デス他
全機種共通システム　実践Small-C講座(3)COMMAND.OBJ2

7月号
特集　超空間美術論
特別付録　DōGA CGAシステム&お試しディスク(5"2HD)

連載　よいこのSX-WINDOW/響子 in CGわ～るど/Z80's Bar
ANOTHER CG WORLD/大人のためのX68000
Computer Music入門/ハード工作/ショートプロ

●試用レポート　V70アクセラレータボード
LIVE in '92　Bye Bye My Love/MATERIAL GIRL/ヴェクザシオン
THE SOFTOUCH　将棋聖天&棋太平68K/シムアース/太閤立志伝
全機種共通システム　実践Small-C講座(4)関数リファレンス

8月号
特集　プログラミング再入門

連載　響子 in CGわ～るど/吾輩はX68000である/よいこのSX-WINDOW
マシン語プログラミング/ハード工作/ANOTHER CG WORLD
大人のためのX68000/Computer Music入門/ショートプロ

●新製品紹介　MATIER,TG100,SOUND SX-68K
LIVE in '92　氷穴/ガラガラヘビがやってくる/風の贈り物
THE SOFTOUCH　三國志III/シムアース/ウルティマVI/バトルテック
全機種共通システム　実践Small-C講座(5)ワイルドカード
グラフィックライブラリGRAPH.LIB

THE SENTINEL

〈対応機種一覧〉●MZ-80K/C/700/1500●MZ-80B/2000●MZ-2500/2861●X1●X1turbo/Z●PC-8001/8801/88●SMC-777/C●PASOPIA/5●PASOPIA 7●FM-7/77/AV●PC-286/386/9801/98●X68000
掲載されたプログラムの利用には各機種用のS-OS"SWORD"システムが必要です。

第124部 MAGIC用モデラ O-EDIT データコンバータ MODCNV

●MAGIC用モデラ

先月の予告どおり，今月はGRAPH.LIBを使ったMAGIC用モデラ「O-EDIT」をお届けします。これは，ワイヤーフレームの物体を作成する支援ツールです。

黒木さんはOh!Xに掲載されたX68000版のMAGIC用モデラを見て，S-OS版の作成を決意したようです。使用感はいたって快適で，解像度の差はあるにせよ，X68000版と比べても，決してひけを取りません。といったらいいすぎですが，実際ユーザーインタフェイス以外はほぼ同等のものを持っているといえます。このプログラムを見れば，ひと目でSLANGとMAGICを統合した，「GRAPH.LIB」の有用性を理解してもらえることでしょう。

ただ残念なことには，先月号の扉の写真にあるポリゴンの描画機能には対応していず，ワイヤーフレームのモデルしか作れません。メモリ容量の関係で削除となってしまったようですが，やる気は失っていないようなのでぜひ期待したいところです。

いままで，MAGICを自在に操れる言語は，アセンブラしかありませんでした。そのため，高級言語指向の人たちには敬遠されてきた面もあるかと思います。しかし，「GRAPH.LIB」の登場により，SLANGでもMAGICを本格的に扱える環境が提供されました。高級言語で扱うことができるようになったMAGICの世界が，どのように展開していくのか，非常に興味深いですね。

●MODCNV

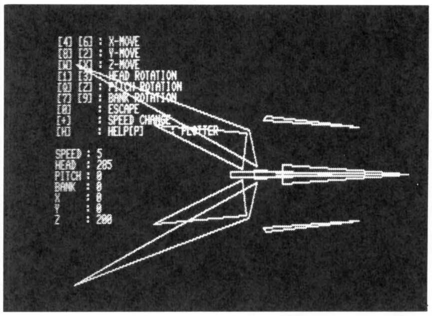
さて，モデラで作った物体データを，そのまま眠らせておく手はありません。そこで2つ目のプログラムとして，MODCNVが用意されています。モデラで作ったデータを，先月号の「GRAPH.LIB」で扱えるような形式に書き出してくれます。

その後は……。もう続ける必要はありませんよね。モデラで作った物体をただぐるぐる回して楽しむのもいいでしょう。しかし，ここまでお膳立てしてもらっておいて，活用しないのはあまりにももったいないことです。使いこなすのは容易ではありませんが，努力したぶんの成果はきっと得られることでしょう。

また，グラフィックを使ったプログラムをこのTHE SENTINELに採用するのは確かに躊躇が伴います。でもスゴイ作品については別です。こちらとしても，よい作品は載せないわけにはいかないのですから。力作をお待ちしています。

●S-OSの系譜(36)

「ELFES」といえばS-OSの歴史に残るシューティングですが，1988年の11月号では，その第3弾「ELFES Ⅳ」が発表されました。3



作目にもかかわらず，名前はなぜか「ELFES Ⅳ」。これは，「ELFES II」との間にもうひとつストーリーが用意される予定だったのです。

このシリーズのサイドストーリーは，一貫して暴走した自動支援戦艦「ELFES」を倒せ，というものです。1，2作目ではあと一歩のところで「ELFES」を逃していました。で，この3作目でも地上要塞となった「ELFES」中枢部にたどり着いたものの……というところで終わってしまい，結局，最終目的が達成されていません。

ゲームシステムもバラエティーに富んでいて，1作目で疑似3D，2作目でサイドビューときて，3作目の「ELFES Ⅳ」ではトップビューの画面を採用。パワーアップあり，オプションありのシューティングであるにもかかわらず，1作目からの高速性はそのままに，回を追うごとに演出は派手に，ボスキャラは大きくなっていきました。このような，次々と新しいものに挑戦しようとする作者の意欲は，並ではありません。

惜しむらくは，記事の中でほのめかされている第4作「ELFES III」が，未発表であること。やはり，いろいろとお忙しくなったのでしょうか。S-OSの歴史に残るほどの名作。このまま完結せずに終わるのは，しのびないものです。

1992■インデックス

■92年1月号
第115部 LINER
■92年2月号
第116部 シミュレーションゲームPOLANYI
■92年3月号
第117部 カードゲームKLONDIKE
■92年4月号
第118部 オプティマイザO80実践Small-C講座(1)
■92年5月号
第119部 COMMAND.OBJ実践Small-C講座(2)
■92年6月号
第120部 COMMAND.OBJ2実践Small-C講座(3)
■92年7月号
第121部 関数リファレンス実践Small-C講座(4)
■92年8月号
第122部 ワイルドカード実践Small-C講座(5)
第123部 グラフィックライブラリ GRAPH.LIB

THE SENTINEL

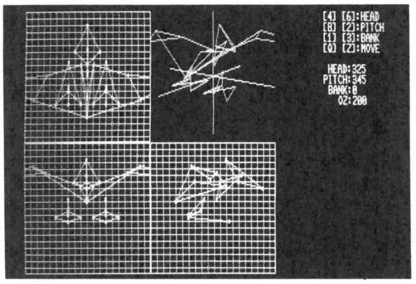
全機種共通
S-OS"SWORD"要

MAGIC用モデラ&データコンバータ
O-EDIT MODCNV

要MAGIC，GRAPH.LIB，NEWファイル入出力ライブラリ

Kuroki Junichi
黒木 淳一

GRAPH.LIBを使ったユーティリティの第1弾として，MAGIC用モデラ「O-EDIT」をお届けします。これで方眼紙から物体データを取らずにすみます。手軽に3D世界を楽しんでください。



今月は予告どおり，MAGIC用モデラ「O-EDIT」を発表します。そして，「O-EDIT」で作成されたデータをSLANGで使えるようなソースファイルにする，コンバータ「MODCNV」も一緒に発表します。

MAGICには，3D物体を扱うための機能が付いています。しかし，簡単に扱うといっても結局人間がデータを作成する必要があります。たとえ方眼紙を使ったとしてもしょせんは2次元の世界，自分の思いどおりのものを作るためには，かなりの慣れが必要になってきます。

そこでこの「O-EDIT」の出番というわけです。2次元と3次元の表示を同時に行いながらエディットできるため，3Dキャラクタを作成する手間がかなり軽減されることでしょう。

入力方法&起動方法

まず，何かのテキストエディタを使ってリスト3を入力してください。入力が終わったら"O-EDIT.SL"としてセーブし，

　]C　O-EDIT.SL

のようにしてSLANGでコンパイルしてください。コンパイルが終わったら，

　]S　3000　終了番地　3000　9000：O-EDIT

のようにセーブします。

なお，このプログラムをコンパイルするときには「GRAPH.LIB」が，実行するときには「MAGIC」が必要です。このプログラムでは「SOROBAN.LIB」を使用していませんので，メモリを余裕を持って確保したい方は「GRAPH.LIB」中の，

　#INCLUDE SOROBAN.LIB

を削除してください。

起動方法は，S-OSのコマンドラインから，

　# O-EDIT［：FILE NAME.MOD］

のようにします。"FILE NAME.MOD"を指定すると，起動時にエディットしたい物体データを読み込んでくれます。もちろん［ ］内のファイルは省略可能です。

機能説明

このプログラムを起動すると図1のような画面になります。基本となるのはこの画面で，操作についてはテンキーまたはカーソルキーを中心にしています。データのセットには"5"，"リターン"キー，キャンセルに"0"，"スペース"キーを使用します。それぞれのメニューで使用するキーが違うこともありますが，画面右のところには常にキー操作が表示されています。わからなくなったら，そちらを参考にしてください。

それでは，メインメニューにある機能を説明していきます。

1)　SET POINT

その名のとおり，物体の頂点をセットします。テンキー（カーソル）の上下左右でX,Y方向の移動，"Q""Z"キーでZ方向の移動ができます。そして，カーソルの移動量を"＋""S"キーで調節することができます（デフォルトは移動量＝5）。思いどおりの場所に移動したら，決定キーを押しましょう。すると，カーソルのあった場所に頂点がセットされます。

2)　SET WIRE

ここでは，1)でセットした頂点を結ぶ線分を引くことができます。メニューを選択すると画面の色が変わり，セットした頂点のひとつが赤くなったと思います。この赤くなっている頂点が，選択しようとしている頂点です。ここでテンキーの上下を押せば，次々と別の点に移動します。線分を結びたい点のひとつにきたら，決定キーを押してください。すると，次の点を要求していきますので同じように頂点を選択して決定キーを押します。このような操作を行うことで2頂点を結ぶ線分が引かれます。

3)　ERASE POINT

このメニューで不要となった頂点を削除します。基本操作は1)と同じです。注意してほしいのは，頂点を削除するとその頂点に結ばれている線分が消去されてしまう点です。一応，消去したい頂点と線分の関係を確認してから実行してください。

4)　ERASE WIRE

すでに描画してある線分の消去を行います。基本操作は2)と同じ，特に注意することはありません。

5)　ROTATION

エディットしている物体の表示角度，位置を調節します。

6)　PALET ON

7)　PALET OFF

パレットを切り替えてグラフィックの表示をON/OFFします。

8)　POINT MOVE

1)で設定した頂点の移動を行います。まず，移動させる頂点を選択してから好みの場所まで移動させてください。部分的な修正をするときに役立つでしょう。

S)　SAVE DATA

L)　LOAD DATA

エディットした物体のセーブ，ロードを

行います。ディレクトリが表示されたあとにファイル名を要求してきますので，拡張子を付けずに入力してください。拡張子は“.MOD”が自動的に付きます。

C)　CLEAR

現在エディットしているデータをクリアします。

R)　ROTATION2

画面が切り替わり，エディットしている物体が画面全体に表示されます。全体像の確認用に使ってください。

M)　MOVE

エディットしている物体の中心座標を変更します。

E)　END

「O-EDIT」を終了します。

H)　HELP

いわゆるヘルプ機能です。「O-EDIT」の基本的な使い方を表示していきます。記憶力に自信のある方は，ソースリスト中のこの部分を入力しなくてもかまいません。

多少，操作にとまどうことがあるかもしれませんが，習うより慣れろ，ということで実際に使ってみてください。

データ構造

次にこの「O-EDIT」で作成されるデータの構造を説明します。前項で述べたとおり，作成されるデータファイル名の拡張子は“.MOD”で，データ構造は表1のとおりとなっています（アドレスはオフセットアドレスです）。

そして，頂点データは［X1］［Y1］［Z1］［X2］［Y2］［Z2］……，線分データは［P1］［P2］［P1］［P2］［P1］［P2］……のように格納されています。セーブされる領域はオフセットアドレス（WORD_DATA［ ］）の0000_H-$0FFF_H$となっています。とりあえず，SLANGで使うためのコンバータは用意しました。アセンブラなどで使いたい人は，これらの情報をもとにコンバータを制作してください。

MODCNV

冒頭で述べたとおりこの「MODCNV」は，「O-EDIT」で作成された3Dキャラクタデータを，SLANGで使用できるようなテキスト形式にコンバートしてくれるものです。ただし，SLANGで使えるといっても「GRAPH.LIB」で使うことを前提としています。なお，コンパイルするためには，1991年9月号で発表された「NEWファイル入出力ライブラリ」が必要ですので注意してください。結構シンプルなプログラムですので，すぐに打ち込めることでしょう。

使い方

このプログラムも「O-EDIT」と同じくSLANGで記述されているので，エディタでリスト4のプログラムを打ち込み，

］C　MODCNV.SL

としてコンパイルしてください。うまくコンパイルできたら，

］S　3000　終了番地　3000　9000：MODCNV

としてセーブしましょう。

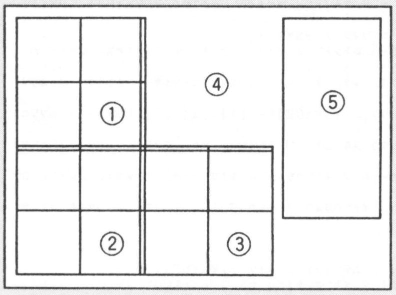
使い方は，S-OSのコマンドラインから，

#L　MODCNV

#J3000

とするだけです。そして，実行後にディレクトリが表示されコンバートするファイル名を聞いてきます。ここで，「O-EDIT」を使って作成したデータファイルを入力してください。すると，入力したファイル名の拡張子を“.SL”にして，SLANGのソース

図1　基本画面



①XZ平面
②XY平面
③YZ平面
④エディット物体表示
⑤メインメニュー

表1　データ構造

オフセット アドレス	
0000_H-$05FF_H$	頂点データ（WORD_DATA［ ］）
0600_H-$07FF_H$	線分データ（BYTE_DATA［ ］）
0800_H	頂点の数（WORD_DATA［ ］）
0802_H	線分の数(WORD_DATA［ ］)

リスト1

```
// MAGIC 3D DATA

CONST PANEL_PMAX= 003,PANEL_WMAX= 003; (*座標、線分の数 *)

ARRAY     PANEL_D[ 003][2]=[                    (* 座標データ *)
          %-00100,%-00100,% 00000,              (* X , Y , Z *)
          %-00100,% 00100,% 00000,              (* X , Y , Z *)
          % 00100,% 00100,% 00000,              (* X , Y , Z *)
          % 00100,%-00100,% 00000],             (* X , Y , Z *)

 BYTE     PANEL_W[ 003][1]=[                    (* 線分  データ *)
           000, 001, 001, 002, 002, 003, 003, 000];

```

リスト2

```

ORG     $9000;

CONST _PLOTSW=0,_THREE=1,_FLOAT=0,_SINCOS=0,_TILESW=0,_GHIN=0;

#INCLUDE GRAPH.LIB

// MAGIC 3D DATA

CONST     PANEL_PMAX= 003,PANEL_WMAX= 003; (* 座標、線分の数 *)

ARRAY     PANEL_D[ 003][2]=[                    (* 座標データ *)
          %-00100,%-00100,% 00000,              (* X , Y , Z *)
          %-00100,% 00100,% 00000,              (* X , Y , Z *)
          % 00100,% 00100,% 00000,              (* X , Y , Z *)
          % 00100,%-00100,% 00000],             (* X , Y , Z *)

 BYTE     PANEL_W[ 003][1]=[                    (* 線分  データ *)
           000, 001, 001, 002, 002, 003, 003, 000];

MAIN()
 VAR I;
 [
    @INIT();                                (* 初期化              *)
    FOR I=0 TO 8 [ _PAR[I] = 0; ]           (* パラメータ  初期化  *)
    @SETOD( PANEL_D , PANEL_PMAX );         (* 座標データ  定義    *)
    @SETWD( PANEL_W , PANEL_WMAX );         (* 線分データ  定義    *)
    _PAR[2] = 5000;                         (* Z = 5 0 0 0         *)
    WHILE( _PAR[2] .>. 0 )          (* Z  がマイナスになるまで *)
    [
        @MAGIC(0);                          (* 2 D  →  3 D        *)
        @MAGIC(1);                          (* 画面表示            *)
        @CRTKN( 7 , 0 , 0);                 (* プレーン切り換え    *)
        @CLS();                             (* 裏プレーン消去      *)
        _PAR[2]=_PAR[2]-15;                 (* Z = Z - 1 5         *)
    ]
 ]
```

形式になったファイルを生成してくれます。出力されたソースは，リスト１のように初期値を設定した配列と頂点数，線分数を設定した定数が書き込まれます。ちなみにリスト１には注釈がありますが，実際には出力されません。

そして，この出力されたソースをプログラムへ組み込んでみたものがリスト２です。このリスト２で注目してほしいのは，26，27行です。@SETOD, @SETWD関数の引数に出力された配列名と，定数名があるのがわかるでしょう。このように記述するだけで，3D物体のキャラクタ定義をすることができるのです。

最後に

以上で説明は終わりです。さすがにマウスが使えないため，ちょっと使いづらいところもありますが，MAGIC対応のソフトを作るときにはきっと役に立つと思います。今回はワイヤーフレームの3D物体のみでしたが，ポリゴンを使えれば完璧。といってもメモリが足りなかったので作成途中で外しちゃいましたけどね。今度は機能を一部削除してでも作ってみたいです。

最後に「MODCNV」のファイルライトルーチンを作成してくれたS-OSユーザーズクラブの森喜一郎さん，ありがとうございました。

リスト3 O-EDIT

```
/*
        << O-EDIT Ver 2.1  1992/06/13  23:32:47 >>

                Programed by KUROKI JUNICHI

                                                        */

ORG      $3000;
OFFSET   $6000;
WORK     $D000;

CONST    _PLOTSW = 1 , _SINCOS = 0 , _THREE = 1 ,
         _TILESW = 1 , _FLOAT = 0  , _GHIN = $A8;

#INCLUDE GRAPH.LIB

ARRAY    WORD    OBJD0[255][2]:$9000,
         BYTE    OBJW0[255][1]:$9600,
         WORD    MSTEP[3] = [%1,%2,%5,%10],
         WORD    LSTEP[5] = [%1,%5,%10,%40,%60,%100],
         WORD    DUMMYP[8] = [%0,%0,%0,%0,%0,%0,%0,%0,%0],
         BYTE    DUMMYW[5] = [0,1,1,2,2,0],
         BYTE    FLNAME[20];

VAR      PMAX:$9800 ,WMAX:$9802 ,PX ,PY ,PZ ,KFLAG ,NOWP ,
         SPEED , FNAD , FNLN , GENX , GENY , GENZ ,
         DTADR = $1F70 , SIZE = $1F72 , EXADR = $1F6E,
         FILE = $1FA3 ,WOPN  = $1FAF , ROPN = $2009 ,
         WRD   = $1FAC , RDD  = $1FA6 , MPOT;

MAIN()
  VAR   KEY , R , FF;
  BEGIN
    GETREG(); FNAD = ^DE; FF = 0;
    START:
    @PALET(0 , 0 , 0 , 0 , 0 , 0 , 0 , 0);
    PMAX  = 0; WMAX = 0; PX   = 0; PY   = 0; PZ   = 0;
    SPEED = 2; NOWP = 0; GENX = 0; GENY = 0; GENZ = 0;
    IF (MEM[FNAD] > 0) AND (FF == 0) [ LOADFILE(FNAD); FF = 1; ]
    FNAD  = &FLNAME;
    FOR R = 0 TO 8 [_PAR[R] = 0;]
    _PAR[_HEAD ]  = 75;
    _PAR[_PITCH]  = 360 - 10;
    _SHEAD        = 15; _SPITCH = 360 - 10; _SBANK = 0;
    _OFSX         = 0;  _OFSY   = 0;        _OFSZ  = 200;
    WSCR(); REDRAW();
    MLOOP1:
    @MODE(2,1); @GRAD(0);
    @WINDOW(  1,  1,  98, 98); @FULL(   1,  1, 198, 98);
    @WINDOW(200,  0, 399, 98); @FULL( 200,  0, 399, 99);
    @WINDOW( 1,101,198,198); @FULL( 1,101,198,198);
    @WINDOW(201,101,398,198); @FULL(201,101,398,198);
    PRINT("¥C");
    LOCATE(60 , 0);     PRINT("Select Menu");
    LOCATE(60 , 1);     PRINT("1) SET   POINT");
    LOCATE(60 , 2);     PRINT("2) SET   WIRE");
    LOCATE(60 , 3);     PRINT("3) ERASE POINT");
    LOCATE(60 , 4);     PRINT("4) ERASE WIRE");
    LOCATE(60 , 5);     PRINT("5) ROTATION");
    LOCATE(60 , 6);     PRINT("6) PALET ON");
    LOCATE(60 , 7);     PRINT("7) PALET OFF");
    LOCATE(60 , 8);     PRINT("8  POINT MOVE");
    LOCATE(60 , 9);     PRINT("S) SAVE DATA");
    LOCATE(60 , 10);    PRINT("L) LOAD DATA");
    LOCATE(60 , 11);    PRINT("C) CLEAR");
    LOCATE(60 , 12);    PRINT("R) ROTATION 2");
    LOCATE(60 , 13);    PRINT("M) MOVE        ______ ");
    LOCATE(60 , 14);    PRINT("E) END       (H:HELP)");
    LOCATE(73 , 15);    PRINT("‾‾‾‾‾‾");
    LOCATE(61 , 16);    PRINT("｢ O-EDIT ｣");
    LOCATE(75 , 17);    PRINT("_");
    LOCATE(61 , 18);    PRINT("OSAKA DENKI TSUSIN");
    LOCATE(61 , 19);    PRINT("DAIGAKU KOUTOU");
    LOCATE(64 , 20);    PRINT("GAKKOU");
    LOCATE(61 , 22);    PRINT("E.D.P.S. bu");
    LOCATE(61 , 23);    PRINT("Junichi Kuroki");
    MLOOP2:
    KEY = INKEY(0);
    IF (KEY == 0) GOTO MLOOP2;
    PRINT("¥C");
    CASE KEY OF [
      '1'    SETP();
      '2'    SETW();
      '3'    ERAP();
      '4'    ERAW();
      '5'    ROTATION();
      '6'    @PALET(0 , 1 , 2 , 3 , 4 , 5 , 6 , 7);
      '7'    @PALET(0 , 0 , 0 , 0 , 0 , 0 , 0 , 0);
      '8'    PMOVE();
      'S'    SAVE();
      'L'    LOAD();
      'E'   [@PALET(0, 0, 0, 0, 0, 0, 0, 0); RETURN; ]
      'C'   [READY(); IF (KFLAG == 1) GOTO START;]
      'R'    LOOK();
      'M'    OMOVE();
      'H'    HELP();
      ]
   GOTO MLOOP1;
  END;

HELP()
  BEGIN
  @PALET(0 , 0 , 0 , 0 , 0 , 0 , 0 , 0); PRINT("¥C");

  PRINT(" HELP MODE¥N¥N¥N"); PK();

  PRINT(" << ﾃﾞｰﾀ ｹｲｼｷ ﾆ ﾂｲﾃ >>¥N¥N");
  PRINT(" ｺﾉ ﾂｰﾙ ﾊ､ MAGIC ﾖｳ ﾉ 3D ｷｬﾗｸﾀ ﾃﾞｰﾀ ｦ ｻｸｾｲ ｽﾙ ﾀﾒﾉ ﾂｰﾙ ﾃﾞｽ｡ ﾃﾞ¥N¥N");
  PRINT("ｰﾀ ｹｲｼｷ ﾊ¥N¥N¥N");
  PRINT("ｵﾌｾｯﾄ ｱﾄﾞﾚｽ 0000-05FF ﾏﾃﾞｶﾞ ｻﾞﾋｮｳ ﾃﾞｰﾀ｡  [ WORD DATA ]¥N¥N");
  PRINT("ｵﾌｾｯﾄ ｱﾄﾞﾚｽ 0600-07FF ﾏﾃﾞｶﾞ ｾﾝﾌﾞﾝ ﾃﾞｰﾀ｡  [ BYTE DATA ]¥N¥N");
  PRINT("ｵﾌｾｯﾄ ｱﾄﾞﾚｽ       0800     ｶﾞ ｻﾞﾋｮｳ ﾉ ｶｽﾞ   [ WORD DATA ]¥N¥N");
  PRINT("ｵﾌｾｯﾄ ｱﾄﾞﾚｽ       0802     ｶﾞ ｾﾝﾌﾞﾝ ﾉ ｶｽﾞ   [ WORD DATA ]¥N¥N");
  PK();
  PRINT("ﾃﾞｨｽｸ ﾆ ｷﾛｸ ｻﾚﾙ ﾘｮｳｲｷ¥N¥N");
  PRINT("ｵﾌｾｯﾄ ｱﾄﾞﾚｽ 0000-0FFF ﾏﾃﾞﾄ ﾅｯﾃ ｵﾘﾏｽ｡¥N¥N¥N");
  PRINT("ｷﾛｸ ｹｲｼｷ¥N¥N");
  PRINT("ｻﾞﾋｮｳ ﾃﾞｰﾀ : [ X1 ][ Y1 ][ Z1 ] [ X2 ][ Y2 ][ Z2 ]...¥N¥N");
  PRINT("ｾﾝﾌﾞﾝ ﾃﾞｰﾀ : [P1][P2] [P1][P2] [P1][P2] [P1][P2]...¥N¥N");
  PRINT("ｻﾞﾋｮｳ ｺｽｳ  : [ PCT ]¥N¥N");
  PRINT("ｾﾝﾌﾞﾝ ｺｽｳ  : [ LCT ]¥N¥N");
  PRINT("*ﾁｭｳｲ¥N¥N");
  PRINT(" ﾃﾞｨｽｸ ﾆ ｾｰﾌﾞ ｽﾙﾄｷ､ ｼﾞﾄﾞｳﾃｷ ﾆ ｶｸﾁｮｳｼ 'MOD' ｶﾞ ﾂｷﾏｽ ﾉﾃﾞ､ ｶｸﾁｮｳ¥N¥N");
  PRINT("ｼ ﾊ ﾂｹﾅｲﾃﾞ ｸﾀﾞｻｲ｡ ﾏﾀ､ ﾛｰﾄﾞ ｽﾙﾄｷﾓ ｼﾞﾄﾞｳﾃｷ ﾆ ｶｸﾁｮｳｼ 'MOD' ｦ ﾂｹﾃ¥N¥N");
  PRINT("ﾌｧｲﾙ ｵｰﾌﾟﾝ ｦ ｽﾙﾉﾃﾞ､ ﾂｹﾅｲﾃﾞ ｸﾀﾞｻｲ｡¥N¥N¥N"); PK();

  PRINT(" << ｶｰｿﾙ ﾆ ﾂｲﾃ >>¥N¥N");
  PRINT(" ｺﾉ ｴﾃﾞｨﾀｰﾊ､ﾎﾄﾝﾄﾞ ﾉ ｿｳｻ ｦ ﾃﾝｷｰ ﾃﾞｵｺﾅｴﾏｽ｡ ｹｯﾃｲ ﾊ [5]ｷｰ､ｷｬﾝｾﾙ ﾊ¥N¥N");
  PRINT("ﾄ ﾅｯﾃ ｵﾘﾏｽ｡ ﾏﾀ､ ﾃﾝｷｰ ﾉ ｶﾜﾘ ﾆ ｶｰｿﾙｷｰ､ [5]ｷｰ ﾉ ｶﾜﾘ ﾆ [RETURN]ｷｰ¥N¥N");
  PRINT("[0]ｷｰ ﾉ ｶﾜﾘ ﾆ [SPACE]ｷｰ [1],[3],[7],[9]ｷｰ ﾉ ｶﾜﾘﾆ [J],[K],[U],¥N¥N");
  PRINT("[I]ｷｰ ｦ ｼﾖｳ ｽﾙ ｺﾄ ｶﾞ ﾃﾞｷﾏｽ｡ ﾏﾀ､ ﾓｸﾃｷ ﾉ ｲﾁ ﾆ ｽﾊﾞﾔｸ ｶｰｿﾙ ｦ ｲﾄﾞｳ¥N¥N");
  PRINT("ｻｾﾙ ﾀﾒ､ ｶｰｿﾙ ﾉ ｲﾄﾞｳﾘｮｳ ﾊ ｶﾅﾘ ｵｵｷｸ ﾅｯﾃﾏｽ｡ ｺﾚﾃﾞﾊ ｺﾏｶﾅ ｷｬﾗｸﾀ ｶﾞ¥N¥N");
  PRINT("ﾂｸﾚﾅｲ ﾉﾃﾞ､ ｲﾄﾞｳﾘｮｳ ﾉ ﾍﾝｶ ｦ [S],[+]ｷｰ ﾃﾞ ｻｾﾙ ｺﾄ ｶﾞ ﾃﾞｷﾏｽ｡¥N¥N¥N");
  PK();

  PRINT(" << ﾌﾞｯﾀｲ ﾉ ｴﾃﾞｨｯﾄ ﾆ ﾂｲﾃ >>¥N¥N");
  PRINT(" ﾏｽﾞ､ﾌﾞｯﾀｲ ｦ ｴｶﾞｸ ﾀﾒﾆﾊ ｾﾝ ｦ ﾋｶﾅｹﾚﾊﾞ ｲｹﾏｾﾝ｡ ﾜｲﾔｰﾌﾚｰﾑ ﾊ ｾﾝ ﾀﾞｹ¥N¥N");
  PRINT("ﾉ ｱﾂﾏﾘ ﾃﾞｽｶﾗ､ ｾﾝ ｦ ﾋｸｺﾄ ﾆ ﾅﾚﾃ ｲﾀﾀﾞｹﾚﾊﾞ ｶﾝﾀﾝ ﾆ ｺﾉ ﾂｰﾙ ｦ ﾂｶｲｺﾅｽ¥N¥N");
  PRINT("ｺﾄ ｶﾞ ﾃﾞｷﾏｽ｡¥N¥N");
  PRINT(" ﾏｽﾞ､ ﾋｷﾀｲ ｾﾝ ﾉ ﾘｮｳﾊｼ ﾆ ﾃﾝ ｦ ｳｯﾃ ｸﾀﾞｻｲ｡ [1] ﾉ SET POINT ﾃﾞｽ｡¥N¥N");
  PRINT("ｶｰｿﾙｷｰ､ﾏﾀﾊ ﾃﾝｷｰ ﾃﾞ ｶｰｿﾙ ｦ ｳｸﾞｶｼ､ ﾃﾝ ｦ [ﾘﾀｰﾝ]､ ﾏｸﾊ [5]ｷｰﾃﾞ ﾃﾝ¥N¥N");
  PRINT("ｦ ｳﾂｺﾄ ｶﾞ ﾃﾞｷﾏｽ｡ ﾓｼ､ ﾏﾁｶﾞｴﾃ ﾍﾝﾅ ｲﾁ ﾆ ﾃﾝ ｦ ｳｯﾃ ｼﾏｯﾀ ﾊﾞｱｲ､ [0]¥N¥N");
  PRINT("ｷｰ ﾏﾀﾊ [ｽﾍﾟｰｽ]ｷｰ ﾃﾞ SET POINT ｦ ﾇｹﾃ [3]ｷｰ ﾃﾞ ERASE POINT ﾆ ﾊ¥N¥N");
  PRINT("ｲﾘ､ [8],[2] ﾏﾀ ﾊ ｶｰｿﾙ ﾃﾞ ｾｯﾄ ｼﾀ ﾃﾝ ｦ ｴﾗﾋﾞ､ ｹｯﾃｲ ｷｰ ﾃﾞ ｻｸｼﾞｮ¥N¥N");
  PK();
  PRINT("ｼﾃｸﾀﾞｻｲ｡ ﾓｼ ﾃﾝ ｦ ｹｽﾉｶﾞ ｲﾔ ﾃﾞｼﾀﾗ､[8]ｷｰ ﾉ POINT MOVE ﾃﾞ, ERASE-¥N¥N");
  PRINT("-POINT ﾉ ﾄｷ ﾄﾞｳﾖｳ ﾆ ﾃﾝ ｦ ｴﾗﾋﾞ､ ﾂｷﾞﾆ ﾓｸﾃｷ ﾉ ｲﾁ ﾆ ｶｰｿﾙ ｦ ｳｺﾞｶｼ¥N¥N");
  PRINT("ｹｯﾃｲ ｷｰ ｦ ｵｼﾃｸﾀﾞｻｲ｡ ｱﾔﾏｯﾃ ｾｯﾄ ｼﾀ ﾃﾝ ｶﾞ ｿﾉ ﾓｸﾃｷ ﾉ ｲﾁ ﾆ ｲﾄﾞｳ ｼ¥N¥N");
  PRINT("ﾏｽ｡¥N¥N");
  PRINT(" ﾌﾀﾂ ﾉ ﾃﾝ ｦ ｳｯﾀﾗ ﾂｷﾞﾊ ｷｬﾝｾﾙ ｷｰ ﾃﾞ ﾒﾆｭｰ ﾆ ﾓﾄﾞﾘ､ [2
```


▶トルクとかの概念を使わずに，よくここまで花瓶の動きをシミュレートできましたね。こういう記事を見ると，僕にもやる気がわいてきます。　佐伯　章(19)東京都

```
] ﾃﾞ SET WI-¥N¥N");
    PRINT("-RE ｦ ｴﾗﾝﾃﾞ ｸﾀﾞｻｲ､ ERASE POINT ﾄ ｵﾅｼﾞ ﾖｳﾆ ﾃﾝ ｦ ｾﾚ
ｸﾄ ﾃﾞｷﾏｽ ﾉﾃﾞ¥N¥N");
    PRINT("ﾃﾝ ｦ ﾌﾀﾂ ｴﾗﾝﾃﾞ ｸﾀﾞｻｲ｡ ｾﾝ ｶﾞ ﾋｹﾏｽ｡ ﾓｼ ﾏﾁｶﾞｴﾃ ｾﾝ ｦ
ﾋｲﾃｼﾏｯﾀﾗ､ ｴ¥N¥N");
    PRINT("ｽｹｰﾌﾟ ｷｰ ﾃﾞ ﾒﾆｭｰ ﾆ ﾓﾄﾞﾘ､ ERASE POINT ﾉ ﾖｳﾘｮｳ ﾃﾞ ｾ
ﾝ ｦ ｻｸｼﾞｮ ｼ¥N¥N");
    PRINT("ﾃｸﾀﾞｻｲ｡¥N¥N¥N"); PK();

    PRINT(" << ﾛｰﾃｰｼｮﾝ ｷﾉｳ ﾆ ﾂｲﾃ >>¥N¥N");
    PRINT(" ｴﾃﾞｨｯﾄ ｼﾃｲﾙ ｻｲﾁｭｳ ﾆ ﾃﾝ ｶﾞ ｵｵｸﾅｯﾀﾘ､ ｳｲﾝﾄﾞｳ ｶﾗ ﾊﾐﾀ
ﾞｽ ﾖｳﾅ ｵｵｷﾅ¥N¥N");
    PRINT("ｷｬﾗｸﾀ ｦ ﾂｸｯﾃ ｲﾙﾄ､ ｶﾅﾘ ﾌﾍﾞﾝ ﾆ ﾅﾘﾏｽ｡ ｿﾝﾅ ﾄｷﾊ､ [5] ﾉ
 ROTATION ｶ,¥N¥N");
    PRINT("[M] ﾉ MOVE ｦ ｴﾗﾝﾃﾞ ｸﾀﾞｻｲ｡ ROTATION ﾃﾞﾊ､ 3D ｳｲﾝﾄﾞｳ
 ﾉ ｷｬﾗｸﾀ ｦ¥N¥N");
    PRINT("ｶｲﾃﾝ ｻｾﾀﾘ､ ｻｲｽﾞ ｦ ﾁｲｻｸ ｻｾﾙ ｺﾄ ｶﾞ ﾃﾞｷﾏｽ｡ MOVE ﾃﾞﾊ
ﾋﾄﾂ ｶｰｿﾙ ｶﾞ¥N¥N");
    PRINT("ﾃﾞﾏｽ ﾉﾃﾞ､ ｽｷ ﾅ ｲﾁ ﾃﾞ ｹｯﾃｲ ｷｰ ｦ ｵｽﾄ､ ｿﾉ ｲﾁ ｦ ﾁｭｳｼﾝ
 ﾄ ｼﾃ､ ｷｬﾗｸ¥N¥N");
    PRINT("ﾀ ﾉ ｻﾞﾋｮｳ ｦ ｻｲﾃｲｷﾞ ｼﾏｽ｡-¥N¥N¥N"); PK();

    PRINT(" << ﾂｸｯﾀ ｷｬﾗｸﾀ ｦ ﾀﾉｼﾐﾀｲ ﾋﾄ ﾉ ﾀﾒﾆ >>¥N¥N");
    PRINT(" ｺﾉ ｴﾃﾞｨﾀ ﾊ ﾂｸｯﾀ ｷｬﾗｸﾀ ｦ ﾅﾆｶ ﾉ ｿﾌﾄｳｴｱ ﾃﾞ ﾂｶｳ ﾏｴ ﾆ
 ﾄﾞﾝﾅ ｶﾝｼﾞ¥N¥N");
    PRINT("ﾃﾞ ﾋｮｳｼﾞ ｻﾚﾙｶ ｶｸﾆﾝ ｽﾙ ｺﾄ ｶﾞ ﾃﾞｷﾏｽ｡ ﾒﾆｭｰﾃﾞ [R] ﾉ R
OTATION 2 ｦ¥N¥N");
    PRINT("ｴﾗﾝﾃﾞ ｸﾀﾞｻｲ｡ ｶﾞﾒﾝ ｲｯﾊﾟｲ ﾆ ﾜｲﾔｰﾌﾚｰﾑ ﾋｮｳｼﾞ ｦ ｼﾃｸﾚﾏｽ
｡ ｶﾞﾒﾝ ﾆ ｿｳ¥N¥N");
    PRINT("ｻ ﾎｳ ｶﾞ ﾋｮｳｼﾞ ｻﾚﾏｽ ﾉﾃﾞ､ ｿﾚ ﾆ ｼﾀｶﾞｯﾃ ｽｷ ﾅ ｲﾁ ｶﾗ ﾐﾃ
ｸﾀﾞｻｲ｡¥N¥N¥N");
    PK();

    PRINT(" << SAVE/LOAD ｷﾉｳ ﾆ ﾂｲﾃ >>¥N¥N");
    PRINT(" ｺﾉ ﾂｰﾙ ﾊ ｻｸｾｲ ｼﾀ ﾃﾞｰﾀ ｦ ﾃﾞｨｽｸ ﾆ ﾎｿﾞﾝ ｽﾙｺﾄ ｶﾞ ﾃﾞｷ
ﾏｽ｡ MENU ﾃﾞ¥N¥N");
    PRINT("[S] ｦ ｾﾝﾀｸ ｽﾙﾄ ﾃﾞｨﾚｸﾄﾘ ｶﾞ ﾋｮｳｼﾞ ｻﾚﾃ､ ﾌｧｲﾙ ﾈｰﾑ ｦ ﾆ
ｭｳﾘｮｸ ｽﾙﾖｳ¥N¥N");
    PRINT("ﾖｳｷｭｳ ｼﾃｷﾏｽ｡ ｺｺﾃﾞ､ ｶｸﾁｮｳｼ ｦ ｲﾚｽﾞ ﾆ 13 ﾓｼﾞ ｲﾅｲ ﾆ ﾌ
ｧｲﾙ ﾈｰﾑ ｦ ﾆ¥N¥N");
    PRINT("ﾕｳﾘｮｸ ｼﾃ ｸﾀﾞｻｲ｡ ｼﾞﾄﾞｳﾃｷ ﾆ ｶｸﾁｮｳｼ 'MOD' ｶﾞ ﾂｷ､ ｼｭｳ
ﾘｮｳ ｼﾏｽ｡¥N¥N");
    PRINT(" LOAD ﾆ ﾂｲﾃ ﾊ SAVE ﾄ ﾄﾞｳﾖｳ ﾉ ｿｳｻ ｦ [L] ｦ ｾﾝﾀｸ ｼﾀ
ｱﾄ ｵｺﾅｯﾃ ｸﾀﾞ¥N¥N");
    PRINT("ｻｲ｡ ｶｸﾁｮｳｼ ﾊ ｲﾘﾏｾﾝ｡¥N¥N¥N"); PK();

#IF (_PLOTSW == 1)

    PRINT(" << ﾌﾟﾛｯﾀ ｼｭﾂﾘｮｸ ｷﾉｳ ﾆ ﾂｲﾃ >>¥N¥N");
    PRINT(" ｺﾉ ﾂｰﾙ ﾃﾞﾊ､ ROTATION 2 ｦ ｾﾝﾀｸ ｼﾃﾙ ﾄｷﾉﾐ ﾌﾟﾛｯﾀ ﾆ ｷ
ｬﾗｸﾀ ｦ ｶｷｺﾑ¥N¥N");
    PRINT("ｺﾄ ｶﾞ ﾃﾞｷﾏｽ｡ ｷｬﾗｸﾀ ｦ ｶｲﾃﾝ ｻｾﾃﾙ ﾄｷ ﾆ ｷﾆ ｲｯﾀ ｱﾝｸﾞﾙ
ﾆ ﾅｯﾀﾗ [P]¥N¥N");
    PRINT("ﾃﾞ ﾌﾟﾛｯﾀ ﾆ ｼｭﾂﾘｮｸ ｼﾃ ｸﾚﾏｽ｡ ｼﾖｳ ﾃﾞｷﾙ ﾌﾟﾛｯﾀ ﾊ ﾌﾟﾘﾝﾀ
 ﾎﾟｰﾄ ﾆ ｾﾂ¥N¥N");
    PRINT("ｿﾞｸ ｶﾉｳ ﾃﾞ､ 'H' , 'J1' , 'M x1 , y1' , 'D x1 , y2
' ｶﾞ ﾂｶｴﾙ ﾓ¥N¥N");
    PRINT("ﾉ ﾃﾞｽ｡¥N¥N¥N"); PK();

#ENDIF

    PRINT(" << ｷﾄﾞｳ ｼﾞ ﾉ ﾃﾞｰﾀ ﾖﾐｺﾐ ﾆ ﾂｲﾃ >>¥N¥N");
    PRINT(" ｺﾉ ﾂｰﾙ ｦ ﾀﾁｱｹﾞﾙ ﾄｷ､ ﾄﾞｳｼﾞ ﾆ ﾃﾞｰﾀ ｦ ﾖﾐｺﾑ ｺﾄ ｶﾞ ﾃﾞ
ｷﾏｽ｡¥N¥N¥N");
    PRINT("# O-EDIT:file name.MOD [CR]¥N¥N¥N");
    PRINT(" ﾄ ｲｳﾌｳ ﾆ ｳﾁ ｺﾝﾃﾞ ｸﾀﾞｻｲ｡ file name ﾉ ﾌﾞﾌﾞﾝ ﾆﾊ､ ﾌｧ
ｲﾙ ﾈｰﾑ ｶﾞ¥N¥N");
    PRINT("ﾊｲﾘﾏｽ｡ ｶｸﾁｮｳｼ 'MOD' ﾊ ｼｮｳﾘｬｸ ﾃﾞｷﾏｾﾝ｡¥N¥N¥N"); PK(
);

    PRINT("[EOD]¥N"); BEEP(); BEEP(); BEEP(); PK(); BEEP();

    PRINT("¥C");
    @PALET(0 , 1 , 2 , 3 , 4 , 5 , 6 , 7);
    END;

PK()
    BEGIN
    WHILE(INKEY(0) == 0) []
    END;

WSCR()
    VAR    I;
    BEGIN
      @INIT();
      @MODE(2,0);
      FOR I = 0 TO 39 [ @LINE(I * 10 , 0 , I*10 , 199); ]
      FOR I = 0 TO 39 [ @LINE(0 , I * 5 , 399 , I * 5); ]
      @GRAD(0);
      @MODE(2 , 0); @FULL(200 , 0 , 399 , 99);
      @MODE(2 , 1); @FULL(200 , 0 , 399 , 99);
      @MODE(2 , 2); @FULL(200 , 0 , 399 , 99);
      @WINDOW( 200 , 0 , 399 , 99);
      @MODE(2 , 0);
      @TLINE(0 - _OFSZ ./. 2 , 0 , 0 , _OFSZ ./. 2 , 0 , 0);
      @TLINE(0 , 0 - _OFSZ ./. 2 , 0 , 0 , _OFSZ ./. 2 , 0);
      @TLINE(0 , 0 , 0 - _OFSZ ./. 2 , 0 , 0 , _OFSZ ./. 2);
      @WINDOW( 0 , 0 , 639 , 199);
      @CBOX( 0 ,   0 , 199 ,  99 , 6);
      @CBOX( 0 , 100 , 199 , 199 , 6);
      @CBOX(200 , 100 , 399 , 199 , 6);
    END;

CORSOR()
    VAR    KEY , IX , IY , IZ;
    BEGIN
      DCSR(2 , 1 , PX , PY , PZ);
      LOCATE(60 ,  7);    PRINT("[4]   [6]:X");
      LOCATE(60 ,  8);    PRINT("[8]   [2]:Y");
      LOCATE(60 ,  9);    PRINT("[Q]   [Z]:Z");
      LOCATE(60 , 10);    PRINT("[+]   [S]:SPEED");
      LOCATE(60 , 11);    PRINT("[5] [RET]:SET");
      LOCATE(60 , 12);    PRINT("[0]   [ ]:ESCAPE");
      CORSOR1:
      KEY = INKEY(0);
      IF (KEY == 0) GOTO CORSOR1;
      IX = PX;
      IY = PY;
      IZ = PZ;
      KEY = KEYCNV(KEY);
      CASE KEY OF [
          '4'     PX = PX - MSTEP[SPEED];
          '6'     PX = PX + MSTEP[SPEED];
          '2'     PY = PY + MSTEP[SPEED];
          '8'     PY = PY - MSTEP[SPEED];
          'Z'     PZ = PZ - MSTEP[SPEED];
          'Q'     PZ = PZ + MSTEP[SPEED];
          '+'     [SPEED++; SPEED = (SPEED MOD 4); ]
          'S'     [SPEED++; SPEED = (SPEED MOD 4); ]
          '5'     [KFLAG = 1; GOTO CORSOR2;        ]
          '0'     [KFLAG = 0; GOTO CORSOR2;        ]
          ]
      DCSR(0 , 1 , IX , IY , IZ);
      DCSR(2 , 1 , PX , PY , PZ);
      LOCATE(60 , 0);
      PRINT("X:");         PRINT(PN$( PX)); PRINT("     ");
      LOCATE(60 , 1);
      PRINT("Y:");         PRINT(PN$(-PY)); PRINT("     ");
      LOCATE(60 , 2);
      PRINT("Z:");         PRINT(PN$( PZ)); PRINT("     ");
      LOCATE(60 , 3);
      PRINT("POINT:"); PRINT(     PMAX); PRINT("     ");
      LOCATE(60 , 4);
      PRINT("WIRE :"); PRINT(     WMAX); PRINT("     ");
      LOCATE(60 , 5);
      PRINT("SPEED:"); PRINT(    SPEED); PRINT("   ");
    GOTO CORSOR1;
      CORSOR2:
    END;

KEYCNV(KEY)
    BEGIN
      CASE  KEY OF
      [
          ' '       KEY = '0';      (*      CANSEL KEY      *)
          $0D       KEY = '5';      (*      SET    KEY      *)
          $1E       KEY = '8';      (*      UP              *)
          $1F       KEY = '2';      (*      DOWN            *)
          $1C       KEY = '6';      (*      RIGHT           *)
          $1D       KEY = '4';      (*      LEFT            *)
          'J'       KEY = '1';      (*      LEFT,DOWN       *)
          'K'       KEY = '3';      (*      RIGHT,DOWN      *)
          'U'       KEY = '7';      (*      LEFT,UP         *)
          'I'       KEY = '9';      (*      RIGHT,UP        *)
      ]
    END(KEY);

DCSR(MM , MS , LX , LY , LZ)
    ARRAY WORD     COR[11]=[ %-2,% 2,% 0,
                             % 2,% 2,% 0,
                             % 2,%-2,% 0,
                             %-2,%-2,% 0],
          WORD     COS[11],
          BYTE      COW[7]=[ 0,1 , 1,2 , 2,3 , 3,0];
    VAR    I;
    BEGIN
      @MODE(MM , MS);
      @WINDOW(  1,   1 , 198, 98);
      @BOX( LX + 98, 49 - LZ ./. 2, LX + 100, 51 - LZ ./. 2)
;
      @WINDOW(  1, 101, 198, 198);
      @BOX( LX + 98, LY ./. 2 + 149, LX + 100, LY ./. 2 + 15
1);
      @WINDOW( 201, 101, 398, 198);
      @BOX( LZ + 298,LY ./. 2 + 149,LZ + 300,LY ./. 2 + 151)
;
      @WINDOW( 201,   1, 398, 98);
      FOR I = 0 TO 3
      [
          COS[I * 3     ]=COR[I * 3     ] + LX;
          COS[I * 3 + 1]=COR[I * 3 + 1] + LY;
          COS[I * 3 + 2]=COR[I * 3 + 2] + LZ;
      ]
      @SETOD(&COS , 3); @SETWD(&COW , 3);
      _PAR[      0] =   _OFSX;
      _PAR[      1] =   _OFSY;
      _PAR[      2] =   _OFSZ;
      _PAR[_HEAD] = _SHEAD;
      _PAR[_PITCH] = _SPITCH;
      _PAR[_BANK] = _SBANK;
      @MAGIC(0); @MAGIC(1);
    END;

OMOVE()
    VAR    I;
    BEGIN
      PX = 0; PY = 0; PZ = 0;
      OMOVE1:
        CORSOR();
        IF (KFLAG == 1) [
          FOR I = 0 TO 255
          [
            OBJD0[I][0] = OBJD0[I][0] - PX;
            OBJD0[I][1] = OBJD0[I][1] - PY;
            OBJD0[I][2] = OBJD0[I][2] - PZ;
          ]
          REDRAW();
          BEEP(); BEEP();
          PZ =    0; PY =    0; PX =    0;
          GENZ = GENZ + PZ; GENY = GENY + PY;
          GENX = GENX + PX;
          GOTO OMOVE1;
        ]
        IF (KFLAG == 0) RETURN;
      GOTO OMOVE1;
    END;

SETP()
```

▶いつもゲームを作ろうと思っただけで実行できなかった僕ですが，8月号の特集を読んで今度こそやってやる，という気にさせていただきました。　　河田　渉(20)北海道

```
  BEGIN
    SETP1:
    CORSOR();
    IF (KFLAG == 1) [
      IF (PMAX < 250) [ DCSR(2 , 2 , PX , PY , PZ);
                        OBJD0[PMAX][0] = PX;
                        OBJD0[PMAX][1] = PY;
                        OBJD0[PMAX][2] = PZ;
                        BEEP(); BEEP();
                        PMAX++;                    ]
                                                   ]
    IF (KFLAG == 0) [ GOTO SETP2; ]
    GOTO SETP1;
    SETP2:
  END;

PMOVE()
  BEGIN
    PMOVE1:
    IF (PMAX == 0) RETURN;
    SELPOT(); BEEP(); BEEP();
    IF (KFLAG == 1) [
      PX = OBJD0[NOWP][0];
      PY = OBJD0[NOWP][1];
      PZ = OBJD0[NOWP][2];
      CORSOR();
      IF (KFLAG == 1) [
        OBJD0[NOWP][0] = PX;
        OBJD0[NOWP][1] = PY;
        OBJD0[NOWP][2] = PZ;
        REDRAW(); BEEP(); BEEP();
      ]
    ]
    IF (KFLAG == 0) RETURN;
    GOTO PMOVE1;
  END;

SETW()
  BEGIN
    SETW1:
    PRINT("¥C");
    LOCATE(60 , 3); PRINT("[8] [2]:SELECT POINT");
    LOCATE(60 , 4); PRINT("     [5]:SET 1st,2nd ");
    LOCATE(60 , 5); PRINT("         POSITION    ");
    LOCATE(60 , 6); PRINT("     [0]:ESCAPE      ");
    LOCATE(60 , 7); PRINT("MAX:"); PRINT(PMAX);
    SELPOT();
      IF (KFLAG == 1) [
          IF (WMAX < 250) [
              OBJW0[WMAX][0] = NOWP; BEEP(); BEEP();
              LOCATE(60,10); PRINT("FIRST:" , NOWP);
              SELPOT();
              IF (KFLAG == 1) [
                  OBJW0[WMAX][1] = NOWP; BEEP(); BEEP();
                  WMAX++;
              ]
          ]
      ]
      IF (KFLAG == 0) GOTO SETW2;
      @MODE(2 , 2);
      DLINE(WMAX - 1);
      GOTO SETW1;
    SETW2:
  END;

SELPOT()
  VAR   KEY , OLDP;
  BEGIN
    IF (NOWP .>. PMAX - 1) NOWP = PMAX - 1;
    @PALET(0 , 1 , 2 , 3 , 4 , 5 , 2 , 7);
    DCSR(0 , 1 , PX , PY , PZ);
    DCSR(2 , 1 , OBJD0[NOWP][0] , OBJD0[NOWP][1] , OBJD0[NOWP][2]);
    SELPOT1:
    KEY = INKEY(0);
    IF (KEY == 0) GOTO SELPOT1;
    OLDP = NOWP;
    KEY = KEYCNV(KEY);
    CASE KEY OF [
        '8'     [IF (NOWP .<. PMAX - 1) NOWP++;]
        '2'     [IF (NOWP .>.         0) NOWP--;]
        '5'     [KFLAG = 1;       GOTO SELPOT2;]
        '0'     [KFLAG = 0;       GOTO SELPOT2;]
        ]
    LOCATE(60 , 0);
    PRINT("POINT NUMBER:" , NOWP , "    ");
    LOCATE(68 , 1);
    PRINT("WIRE:"         , WMAX , "    ");
    DCSR(0 , 1 , OBJD0[OLDP][0] , OBJD0[OLDP][1] , OBJD0[OLDP][2]);
    DCSR(2 , 1 , OBJD0[NOWP][0] , OBJD0[NOWP][1] , OBJD0[NOWP][2]);
    GOTO SELPOT1;
    SELPOT2:
    DCSR(0 , 1 , OBJD0[OLDP][0] , OBJD0[OLDP][1] , OBJD0[OLDP][2]);
    @PALET(0 , 1 , 2 , 3 , 4 , 5 , 6 , 7);
  END;

SAVE()
  VAR   PARA;
  BEGIN
    PARA = GETFNAME();
    IF (PARA == -1) RETURN;
    MEMW[DTADR] = 0; MEMW[SIZE] = $1000; MEMW[EXADR] = 0;
    ^A = 1; ^DE = FNAD; CALL(FILE);        (* FILE SET  *)
    CALL(WOPN);                            (* FILE OPEN *)
    GETREG(); IF (^CARRY == 1) RETURN;
    MEMW[DTADR] = $9000;
    CALL(WRD);                       (* SAVE       *)
  END;

LOAD()
  VAR   PARA;
  BEGIN
    PARA = GETFNAME();
    IF (PARA == -1) RETURN;
    LOADFILE(FNAD);
    REDRAW();
  END;

GETFNAME()
  VAR   PARA , I;
  BEGIN
    @PALET(0 , 0 , 0 , 0 , 0 , 0 , 0 , 0);
    CALL($2006);
    PRINT("INPUT FILE NAME¥N");
    PARA  = LINPUT(FNAD , 18);
    FOR I = 0 TO 20
    [
      IF (FLNAME[I] == 0)
      [
        FLNAME[I    ] = '.';
        FLNAME[I + 1] = 'M';
        FLNAME[I + 2] = 'O';
        FLNAME[I + 3] = 'D';
        FLNAME[I + 4] =   0;
        I = 20;
      ]
    ]
    @PALET(0 , 1 , 2 , 3 , 4 , 5 , 6 , 7);
  END(PARA);

LOADFILE(FADR)
  BEGIN
    ^A = 1; ^DE = FADR; CALL(FILE);        (* FILE SET  *)
    CALL(ROPN);                            (* FILE OPEN *)
    MEMW[DTADR] = $9000;
    GETREG(); IF (^CARRY == 1) RETURN;
    CALL(RDD);                             (* LOAD       *)
  END;

DLINE(POINT)
  VAR    X1 , X2 , Y1 , Y2 , Z1 , Z2 , I;
  BEGIN
    @WINDOW(200 , 0 , 399 , 99);
    _PAR[     0] = _OFSX;
    _PAR[     1] = _OFSY;
    _PAR[     2] = _OFSZ;
    _PAR[_HEAD] = _SHEAD;
    _PAR[_PITCH] = _SPITCH;
    _PAR[_BANK] = _SBANK;
    FOR I = 0 TO 5
    [
      DUMMYP[I] = OBJD0[OBJW0[POINT][I/3]][(I MOD 3)];
    ]
    @SETOD(&DUMMYP , 1); @SETWD(&DUMMYW , 0);
    @MAGIC(0); @MAGIC(1);
    X1 = DUMMYP[0]; Y1 = DUMMYP[1]; Z1 = DUMMYP[2];
    X2 = DUMMYP[3]; Y2 = DUMMYP[4]; Z2 = DUMMYP[5];
    @WINDOW(  1,  1,198, 98);
    @LINE(X1+ 99, 50-Z1./.2,X2+ 99, 50-Z2./.2);
    @WINDOW(  1,101,198,198);
    @LINE(X1+ 99,Y1./.2+150,X2+ 99,Y2./.2+150);
    @WINDOW(201,101,398,198);
    @LINE(Z1+299,Y1./.2+150,Z2+299,Y2./.2+150);
  END;

REDRAW()
  VAR    I;
  BEGIN
    _GMASK = 6;
    @WINDOW(  1 ,   1 , 198 ,  98);
    @CFULL(   1 ,   1 , 198 ,  98 , 0,0,0);
    @WINDOW(  1 , 101 , 198 , 198);
    @CFULL(   1 , 101 , 198 , 198 , 0,0,0);
    @WINDOW(201 , 101 , 398 , 198);
    @CFULL( 201 , 101 , 398 , 198 , 0,0,0);
    _GMASK = 7;
    @WINDOW(200 , 0 , 399 , 99);
    @CFULL(  200 , 0 , 399 , 99 , 0 , 0 , 0);
    @MODE(2 , 0);
    @TLINE(0 - _OFSZ ./. 2 , 0 , 0 , _OFSZ ./. 2 , 0 , 0);
    @TLINE(0 , 0 - _OFSZ ./. 2 , 0 , 0 , _OFSZ ./. 2 , 0);
    @TLINE(0 , 0 , 0 - _OFSZ ./. 2 , 0 , 0 , _OFSZ ./. 2);
    IF (PMAX == 0) GOTO REDRAW2;
    FOR I = 0 TO PMAX - 1
    [
      DCSR(2, 2, OBJD0[I][0], OBJD0[I][1], OBJD0[I][2]);
    ]
    REDRAW2:
    IF (WMAX == 0) GOTO REDRAW1;
    FOR I = 0 TO WMAX - 1 [ DLINE(I); ]
    REDRAW1:
  END;

ROTATION()
 VAR   I,KEY;
 BEGIN
   @GRAD(0);
   LOCATE(60 , 0); PRINT("[4] [6]:HEAD");
   LOCATE(60 , 1); PRINT("[8] [2]:PITCH");
   LOCATE(60 , 2); PRINT("[1] [3]:BANK");
   LOCATE(60 , 3); PRINT("[Q] [Z]:MOVE");
   ROT1:
     LOCATE(60 , 5);
     PRINT(" HEAD:", _SHEAD); PRINT("   ");
     LOCATE(60 , 6);
     PRINT("PITCH:",_SPITCH); PRINT("   ");
     LOCATE(60 , 7);
     PRINT(" BANK:", _SBANK); PRINT("   ");
     LOCATE(60 , 8);
     PRINT("   OZ:",  _OFSZ); PRINT("   ");
     ROT2:
       KEY = INKEY(0);
       IF (KEY == 0) GOTO ROT2;
       @MODE(2 , 0);
       @WINDOW(200 , 0 , 399 , 99);
```

▶エサを付け忘れたゴキブリホイホイに，５匹ものゴキブリが捕まっていた。こいつら何を考えて生きているのだろう。　伊藤　直也(21)静岡県

```
        KEY = KEYCNV(KEY);
        CASE KEY OF [
                $1E     KEY = '8';
                $1F     KEY = '2';
                $1D     KEY = '4';
                $1C     KEY = '6';
        ]
        CASE KEY OF [
         '4'  [_SHEAD=_SHEAD-5;
               IF ( _SHEAD.<.0) _SHEAD = _SHEAD+360;     ]
         '6'  [_SHEAD =_SHEAD+5; _SHEAD =(_SHEAD MOD 360); ]
         '8'  [_SPITCH=_SPITCH+5; _SPITCH=(_SPITCH MOD 360);
]
         '2'  [_SPITCH=_SPITCH-5;
                IF (_SPITCH.<.0) _SPITCH= _SPITCH+360; ]
         '3'  [_SBANK= _SBANK-5;
                IF ( _SBANK.<.0) _SBANK = _SBANK+360;       ]
         '1'  [_SBANK= _SBANK+5; _SBANK =(_SBANK MOD 360); ]
         'Q'  [IF (_OFSZ<1000) _OFSZ=_OFSZ+20;]
         'Z'  [IF (_OFSZ>0)     _OFSZ=_OFSZ-20;]
         '0'  GOTO ROT3;
        ]
          _PAR[ _HEAD] = _SHEAD;
          _PAR[_PITCH] = _SPITCH;
          _PAR[ _BANK] = _SBANK;
          _PAR[     2] = _OFSZ;
          @FULL(200 , 0 , 399 , 99);
          @TLINE(0-_OFSZ./.2,0,0,_OFSZ./.2,0,0);
          @TLINE(0,0-_OFSZ./.2,0,0,_OFSZ./.2,0);
          @TLINE(0,0,0-_OFSZ./.2,0,0,_OFSZ./.2);
          IF ((PMAX > 0) AND (WMAX > 0)) [
              @MODE(2 , 2); @FULL(200 , 0 , 399 , 99);
              @SETOD(&OBJD0 , PMAX     );
              @SETWD(&OBJW0 , WMAX - 1);
              @MAGIC(0);
              @FULL(200 , 0 , 399 , 99);
              @MAGIC(1);
              @MODE(2 , 0);
          ]
          GOTO ROT1;
      ROT3:
      REDRAW();
    END;

ERASEW(NUMBER)
  VAR   I;
  BEGIN
    @MODE(0 , 2); DLINE(NUMBER);
    @MODE(0 , 1); DLINE(NUMBER);
    FOR I = NUMBER TO WMAX [
        OBJW0[I][0] = OBJW0[I + 1][0];
        OBJW0[I][1] = OBJW0[I + 1][1];
    ]
    WMAX--; BEEP(); BEEP();
    IF (WMAX .<. 0) WMAX = 0;
  END;

ERASEP(NUMBER)
  VAR   I , F;
  BEGIN
    DCSR(0 , 2 , OBJD0[NUMBER][0] , OBJD0[NUMBER][1] , OBJ
D0[NUMBER][2]);
    FOR I = NUMBER TO PMAX - 1 [
        OBJD0[I][0] = OBJD0[I + 1][0];
        OBJD0[I][1] = OBJD0[I + 1][1];
        OBJD0[I][2] = OBJD0[I + 1][2];
    ]
    PMAX--; BEEP(); BEEP();
    F=0;
    IF (WMAX .>. 0) [
        FOR I = 0 TO WMAX [
            IF (OBJW0[I][0] == MEM[&NUMBER]) OR
               (OBJW0[I][1] == MEM[&NUMBER])
            [ ERASEW(I); F = 1; ]
        ]
    ]
    IF (WMAX .>. 0)[
      FOR I = 0 TO WMAX [
       IF (OBJW0[I][0] .>. MEM[&NUMBER]) OBJW0[I][0]--;
       IF (OBJW0[I][1] .>. MEM[&NUMBER]) OBJW0[I][1]--;
      ]
    ]
    IF (F == 1) REDRAW();
  END;

ERAW()
  VAR   KEY , NUMBER;
  BEGIN
    NUMBER = 0;
    @PALET(0 , 1 , 2 , 3 , 4  , 5 , 2 , 7);
    IF (WMAX == 0) GOTO ERAW3;
    LOCATE(60 , 0); PRINT("[8] [2]:SELECT");
    LOCATE(60 , 1); PRINT("    [5]:ERASE WIRE");
    LOCATE(60 , 2); PRINT("    [0]:ESCAPE");
    ERAW1:
    LOCATE(60 , 4); PRINT("   PMAX:",   PMAX,".    ");
    LOCATE(60 , 5); PRINT("   WMAX:",   WMAX,".    ");
    LOCATE(60 , 6); PRINT("  ERASE:", NUMBER,".    ");
    KEY=INKEY(0);
    IF (KEY == 0) GOTO ERAW1;
    @MODE(0 , 1); DLINE(NUMBER);
    KEY = KEYCNV(KEY);
    CASE KEY OF
    [
      '2'   [IF (NUMBER >         0) NUMBER--;        ]
      '8'   [IF (NUMBER < WMAX - 1) NUMBER++;         ]
      '5'   [ERASEW(NUMBER);
             IF (NUMBER > WMAX - 1) NUMBER = WMAX - 1;]
      '0'   [GOTO ERAW2;]
    ]
    IF (WMAX == 0) GOTO ERAW2;
    @MODE(2 , 1); DLINE(NUMBER);
    GOTO ERAW1;
    ERAW2:
    REDRAW();
    ERAW3:
    @PALET(0 , 1 , 2 , 3 , 4 , 5 , 6 , 7);
  END;

ERAP()
  VAR   KEY , NUMBER;
  BEGIN
    IF (PMAX == 0) GOTO ERAP3;
    @PALET(0 , 1 , 2 , 3 , 4 , 5 , 2 , 7);
    NUMBER = 0;
    LOCATE(60 , 0); PRINT("[8] [2]:SELECT");
    LOCATE(60 , 1); PRINT("    [5]:ERASE WIRE");
    LOCATE(60 , 2); PRINT("    [0]:ESCAPE");
    ERAP1:
    LOCATE(60 , 4); PRINT("   PMAX:",  PMAX,".    ");
    LOCATE(60 , 5); PRINT("   WMAX:",  WMAX,".    ");
    LOCATE(60 , 6); PRINT("  ERASE:",NUMBER,".    ");
    KEY = INKEY(0);
    IF (KEY == 0) GOTO ERAP1;
    DCSR(0 , 1 , OBJD0[NUMBER][0] , OBJD0[NUMBER][1] , OBJ
D0[NUMBER][2]);
    KEY = KEYCNV(KEY);
    CASE KEY OF
    [
      '2'   [IF (NUMBER > 0)         NUMBER--;           ]
      '8'   [IF (NUMBER < PMAX - 1) NUMBER++;            ]
      '5'   [ERASEP(NUMBER);
             IF (NUMBER > PMAX - 1) NUMBER = PMAX - 1;]
      '0'   [GOTO ERAP2;]
    ]
    IF (PMAX == 0) GOTO ERAP3;
    DCSR(2 , 1 , OBJD0[NUMBER][0] , OBJD0[NUMBER][1] , OBJ
D0[NUMBER][2]);
    GOTO ERAP1;
    ERAP2:
    REDRAW();
    ERAP3:
    @PALET(0 , 1 , 2 , 3 , 4 , 5 , 6 , 7);
  END;

READY()
  VAR   KEY;
  BEGIN
    LOCATE(60 , 15); PRINT("OK ?(Y/N)");
    READY1:
    KEY = INKEY(0);
    IF (KEY == 0) GOTO READY1;
    CASE KEY OF
    [
        'Y'     KFLAG = 1;
        'N'     KFLAG = 0;
        OTHERS  GOTO READY1;
    ]
    PRINT("¥C");
  END;

LOOK()
  VAR   I , KEY , X1 , Y1 , X , Y , X2 , Y2;
  BEGIN
    IF (PMAX == 0) OR (WMAX == 0) RETURN;
    MPOT = 0;
    FOR I = 0 TO 8 [ _PAR[I] = 0; ]
    @INIT();
    @MODE(2 , 0);
    @SETOD(&OBJD0 , PMAX);
    @SETWD(&OBJW0 , WMAX - 1);
    @CRTKN(7 , 0 , 0); @CLS(); @MAGIC(0); @MAGIC(1);
    @CRTKN(7 , 0 , 0); @CLS();
    _PAR[2]=0;
    _PAR[1]=0;
    GAID();
    LOOP:
        KINP();
        @MAGIC(0); @MAGIC(1);
        @CRTKN(7 , 0 , 0);
        @CLS();
        IF (KFLAG == 1) GOTO LEND1;
        LOCATE(0 , 10);
        PRINT("SPEED  : " , MPOT                   ,         "¥N");
        PRINT("HEAD   : " , PN$(_PAR[_HEAD])  ,  "       ¥N");
        PRINT("PITCH  : " , PN$(_PAR[_PITCH]) ,  "       ¥N");
        PRINT("BANK   : " , PN$(_PAR[_BANK])  ,  "       ¥N");
        PRINT("X      : " , PN$(_PAR[0])      ,  "       ¥N");
        PRINT("Y      : " , PN$(_PAR[1])      ,  "       ¥N");
        PRINT("Z      : " , PN$(_PAR[2])      ,  "       ¥N");
    GOTO LOOP;
    LEND1:
    WSCR(); REDRAW();
  END;

GAID()
  BEGIN
    PRINT("[4] [6] : X-MOVE¥N");
    PRINT("[8] [2] : Y-MOVE¥N");
    PRINT("[W] [X] : Z-MOVE¥N");
    PRINT("[1] [3] : HEAD ROTATION¥N");
    PRINT("[Q] [Z] : PITCH ROTATION¥N");
    PRINT("[7] [9] : BANK ROTATION¥N");
    PRINT("[0]     : ESCAPE¥N");
    PRINT("[+]     : SPEED CHANGE¥N");
    PRINT("[H]     : HELP");

#IF (_PLOTSW == 1)

    PRINT("[P]     : PLOTTER");

#ENDIF

  END;

KINP()
  VAR   KEY;
  BEGIN
```

▶ 8月号96ページの話に出ていた「先生とおぼしきオヤジ」は，「シミュレーション」のことを「シュミレーション」といっていた。　　加藤　安弘(17)滋賀県

```
        KEY   = 0; WHILE(KEY == 0) KEY = INKEY(0);
        KFLAG = 0;
        KEY = KEYCNV(KEY);
        CASE KEY OF [
          'H'    [ HELP(); GAID(); ]
          '4'      _PAR[0] = _PAR[0] - LSTEP[MPOT];
          '6'      _PAR[0] = _PAR[0] + LSTEP[MPOT];
          '8'      _PAR[1] = _PAR[1] + LSTEP[MPOT];
          '2'      _PAR[1] = _PAR[1] - LSTEP[MPOT];
          'W'      _PAR[2] = _PAR[2] - LSTEP[MPOT];
          'X'      _PAR[2] = _PAR[2] + LSTEP[MPOT];
          '0'     KFLAG   = 1;
          '1'    [ _PAR[_HEAD] =_PAR[_HEAD]-5;
                   IF (_PAR[_HEAD].<.0)
                   _PAR[_HEAD] = _PAR[_HEAD] +360;       ]
          '3'    [ _PAR[_HEAD] = _PAR[_HEAD] +5;
                   _PAR[_HEAD] =(_PAR[_HEAD] MOD 360);   ]
          '7'    [ _PAR[_BANK] = _PAR[_BANK] +5;
                   _PAR[_BANK] =(_PAR[_BANK] MOD 360);   ]
          '9'    [ _PAR[_BANK] = _PAR[_BANK] -5;
                   IF (_PAR[_BANK].<.0)
                   _PAR[_BANK] = _PAR[_BANK] +360;       ]
          'Q'    [ _PAR[_PITCH]= _PAR[_PITCH]+5;
                   _PAR[_PITCH]=(_PAR[_PITCH] MOD 360);  ]
          'Z'    [ _PAR[_PITCH]= _PAR[_PITCH]-5;
                   IF (_PAR[_PITCH].<.0)
                   _PAR[_PITCH]= _PAR[_PITCH]+360;       ]

#IF (_PLOTSW == 1)
          'P'      @PLOT();

#ENDIF

          'S'    [ MPOT++; MPOT = (MPOT MOD 6);]
          '+'    [ MPOT++; MPOT = (MPOT MOD 6);]
        ]
END;
```

リスト4 MODCNV

```
/**********************************************************
***                                                    ***
**        [ MAGIC DATA ] => [ TEXT DATA ]   Convert     **
**                                                      **
**                MM MM  OOO  DDDD   CCCC NN  N V   V   **
**                M M M O   O D   D C     N N N  V V    **
**                M   M  OOO  DDDD   CCCC N  NN   V     **
**                                                      **
**       Program     Mori Kiichiro   Kuroki Junichi     **
***                                                    ***
**********************************************************/

ORG     $3000;
OFFSET  $6000;
WORK    $D000;

ARRAY   WORD    OBJD0[255][2]:$9000,
        BYTE    OBJW0[255][1]:$9600,
        BYTE          DEC2[4] = [  " 000",0],
        BYTE          DEC4[6] = [" 00000",0],
        BYTE        FLNAME[20],
        BYTE         LABEL[20];

VAR     PMAX:$9800 , WMAX:$9802 ,
        FNAD , FNLN ,
        DTADR = $1F70 , SIZE = $1F72 , EXADR = $1F6E , FILE = $1FA3 ,
        WOPN  = $1FAF , ROPN = $2009 , WRD   = $1FAC , RDD  = $1FA6 ;

MAIN()
  VAR   I,J,CT,PARA;
   BEGIN
    PRINT("¥C <<<< MODCNV V1.00 >>>>¥N");
    FNAD = FLNAME;
    PARA = GETFNAME();
    IF (PARA == -1) RETURN;
    LOADFILE(FNAD);
    FOR I=0 TO 20 [
      IF (FLNAME[I] == '.') [
        FLNAME[I + 1] = 'S';
        FLNAME[I + 2] = 'L';
        FLNAME[I + 3] =  0 ;
        I = 20;
      ]
    ]
/*
** ﾌｧｲﾙ ｶｷｺﾐ
*/
     FOPEN(0,FLNAME,3);
     FPRINT(0,"// MAGIC 3D DATA¥N¥N");
     FPRINT(0,"CONST                 ");

     FPRINT(0,LABEL);
     FPRINT(0,"_PMAX=");
     FPRINT(0,DEC2CNV(PMAX-1));
     FPRINT(0,",");

     FPRINT(0,LABEL);
     FPRINT(0,"_WMAX=");
     FPRINT(0,DEC2CNV(WMAX-1));
     FPRINT(0,";¥N¥N");
/*
** ｻﾞﾋｮｳ ﾃﾞｰﾀ ｾｲｾｲ
*/
     FPRINT(0,"ARRAY                 ");
     FPRINT(0,LABEL);
     FPRINT(0,"_D[");
     FPRINT(0,DEC2CNV(PMAX-1));
     FPRINT(0,"][2]=[¥N");
     FOR I=0 TO PMAX-2 [
       FPRINT(0,"                    ");
       FOR J=0 TO 2 [
         FPRINT(0," %");
         FPRINT(0,DEC4CNV(OBJD0[I][J]));
         FPRINT(0,",");
       ]
       FPRINT(0,"¥N");
     ]
/*
** ﾗｽﾄ 1 ｷﾞｮｳ
*/
     FPRINT(0,"                    ");
     FPRINT(0," %");
     FPRINT(0,DEC4CNV(OBJD0[I][0]));
     FPRINT(0,",");
     FPRINT(0," %");
     FPRINT(0,DEC4CNV(OBJD0[I][1]));
     FPRINT(0,",");
     FPRINT(0," %");
     FPRINT(0,DEC4CNV(OBJD0[I][2]));
     FPRINT(0,"],¥N¥N");
/*
** ｾﾝﾌﾞﾝ ﾃﾞｰﾀ ｾｲｾｲ
*/
     FPRINT(0,"           BYTE    ");
     FPRINT(0,LABEL);
     FPRINT(0,"_W[");
     FPRINT(0,DEC2CNV(WMAX-1));
     FPRINT(0,"][1]=[¥N");
     CT=0;
     FPRINT(0,"                    ");
     FOR I=0 TO WMAX-2 [
       FOR J=0 TO 1 [
         FPRINT(0,DEC2CNV(OBJW0[I][J]));
         FPRINT(0,",");
       ]
       IF (++CT==5) [ FPRINT(0,"¥N                    "); CT=0; ]
     ]
/*
** ﾗｽﾄ 1 ｷﾞｮｳ
*/
     FPRINT(0,DEC2CNV(OBJW0[I][0]));
     FPRINT(0,",");
     FPRINT(0,DEC2CNV(OBJW0[I][1]));
     FPRINT(0,"];¥N");
     FPUTC(0,0);
     FCLOSE(0);
 END;

GETFNAME()
  VAR   PARA , I;
  BEGIN
    CALL($2006);
    PRINT("INPUT FILE NAME¥N");
    PARA  = LINPUT(FNAD , 18);
    FOR I = 0 TO 20  LABEL[I]=FLNAME[I];
    FOR I = 0 TO 20
    [
      IF (FLNAME[I] == 0)
      [
        FLNAME[I    ] = '.';
        FLNAME[I + 1] = 'M';
        FLNAME[I + 2] = 'O';
        FLNAME[I + 3] = 'D';
        FLNAME[I + 4] =   0;
        I = 20;
      ]
    ]
  END(PARA);

LOADFILE(FADR)
  BEGIN
    ^A = 1; ^DE = FADR; CALL(FILE);           (* FILE SET  *)
    CALL(ROPN);                               (* FILE OPEN *)
    MEMW[DTADR] = $9000;
    GETREG(); IF (^CARRY == 1) RETURN;
    CALL(RDD);                                (* LOAD      *)
  END;

FPRINT(FP,BYTE BUF[])
  VAR I;
  BEGIN
    I=0; WHILE ( BUF[I]>0 )
    [
        PRINT(STR$(BUF[I],1));
        FPUTC(FP,BUF[I++]);
        IF (INKEY(0)>0) PK();
    ]
  END;

DEC2CNV(D2)
  BEGIN
    IF (D2 .<. 0) DEC2[0] = '-'; ELSE DEC2[0] = ' ';
    D2 = ABS(D2);
    DEC2[3] = '0'+D2-(D2 / 10 * 10); D2 = D2 / 10;
    DEC2[2] = '0'+D2-(D2 / 10 * 10); D2 = D2 / 10;
    DEC2[1] = '0'+D2-(D2 / 10 * 10);
  END(DEC2);

DEC4CNV(D4)
  BEGIN
    IF (D4 .<. 0) DEC4[0] = '-'; ELSE DEC4[0] = ' ';
    D4 = ABS(D4);
    DEC4[5] = '0'+D4-(D4 / 10 * 10); D4 = D4 / 10;
    DEC4[4] = '0'+D4-(D4 / 10 * 10); D4 = D4 / 10;
    DEC4[3] = '0'+D4-(D4 / 10 * 10); D4 = D4 / 10;
    DEC4[2] = '0'+D4-(D4 / 10 * 10); D4 = D4 / 10;
    DEC4[1] = '0'+D4-(D4 / 10 * 10);
  END(DEC4);

PK()
  BEGIN
    WHILE(INKEY(0)>0)  []
    WHILE(INKEY(0)==0) []
    WHILE(INKEY(0)>0)  []
  END;
```

Z'sSTAFF&Z's-EX用外部ファイル

ジャギー除去に挑戦

Miki Tokutaka 御木 徳高

プログラム原作 佐藤 正春

Z's-EX&Z'sSTAFF ver.3.0兼用の外部ファイルとして，ジャギー除去フィルタを発表します。このフィルタは画面上のドット解像度不足によるギザギザを目立たなくする手軽なツールです。

ジャギー除去とは

今回の外部プログラムは佐藤正春さんからの投稿による「ジャギー除去に挑戦」です。ただし，投稿していただいたのがZ's-EX専用でしたので，Z'sSTAFF ver3.0でも利用できるように手を加えました。Z'sSTAFF対応にしたほかは必要以上に手を加えるのを避けました（ただ面倒だっただけという話もある）。

Z'sSTAFFなどで絵を描いたことのある人ならわかると思いますが，ただ単に画面に線を引こうと思っても（水平や垂直線ならともかく），決して綺麗に引くことはできません。どうやっても階段状になってしまいます。

これは画面を構成するピクセルが縦横に碁盤の目のように並んでおり，そのピクセルが光っているかどうかで直線を表している以上，しかたのないものです。ピクセルを増やせば，見た目はある程度改善されますが，なによりピクセルの数はハードにより制約され，そう簡単に増やせるものではありません。

では，どうすればいいのでしょう。どうやら「ピクセルが光っているかどうか」が怪しそうです。これを「ピクセルがどの程度光っているか」としたらどうでしょう。いま，仮に図1のような線を引きたいとします。前者の方法では左下のような階段になってしまいますが，後者の方法ではどうでしょう。任意のピクセルを正方形（または長方形）と考えて，その正方形の中を直線が通った割合で明るさを決めるようにします(図1右下)。そうすることによって視覚的にはごまかすことができます。

幸いX68000では同時に65536色表示できますので（今回は輝度ビットを使わないので32768色），中間調表示はお手のものです。これをアンチエリアシングといい，レイトレーシングなどの3Dグラフィックではよく用いられる技法です。ただし，3D流の方法ではすでに描かれたビットマップデータにアンチエリアシングをかけることはできませんので，ドットパターンから比率を適度に認識して，疑似的にアンチエリアシングをかけてしまおうというのが今回のジャギー除去の方法です。

比率認識のアルゴリズム

とりあえず，階段状になっている部分を認識する必要がありますので，2×2ドットが図3のようなパターンかどうか片っ端から調べていきます。①のようなパターンはどうしようもないので次のループに移るとして，対角が同色のときを抽出していきます。次に上下左右に並んでいる同色のドット数を数えます。

たとえば，上方向には図4でA1がａかつB1がａでないとき，まず1カウントします。さらにA2がａかつB2がａでないとき，A3がａかつB3がaでないとき……といったようにカウントしていき，条件が満たされないときカウントを中止します。図4の場合ではカウントは3となります。

図1 ライン対比

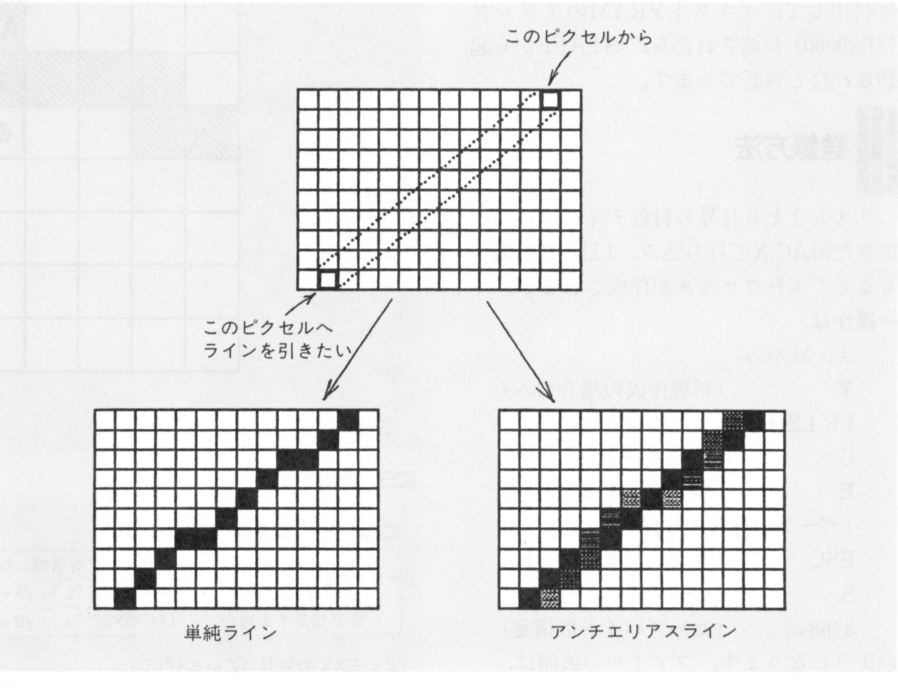
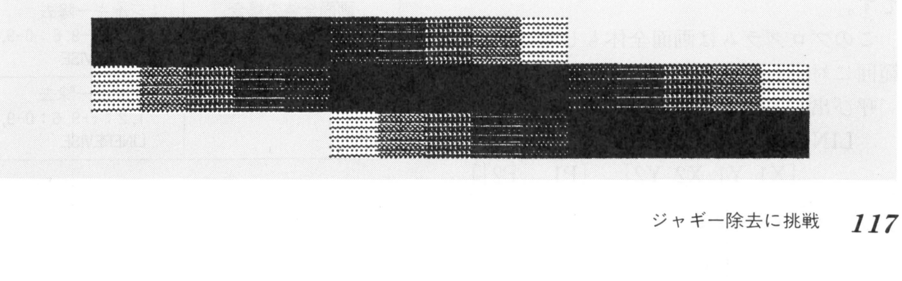


図2 アンチエリアス拡大図



そのようにして，bの上何ドットにaをにじませるかを決めていきます。比率はパラメータで与えられ，b・B1・B2・B3となるにしたがって，比率を少なくします。このように疑似的に色境界のベクトルを拾って，アンチエリアスを下・左右方向にも繰り返してかけていくわけです。

あと，終了時には，Z's STAFFから呼び出された場合のみ，書き換えた部分をテキストVRAMに転送しています。以前も解説したとおり，Z's STAFF側のテキストVRAM→G-RAM転送が無駄になってしまいますが，やむを得ないところでしょう。外部プログラムの終了コードで「転送を行わない」ようにできればいいのですけれどね。

ちなみにZ'sSTAFFは終了コードによって以下のようなメッセージを表示します。

0…正常終了（なにも表示しない）
1…指定のコマンドは実行できません
2以上…コマンド行に間違いがあります

Z'sSTAFFから呼ばれたかどうかは，裏アドレスで判別しています。7月号でも述べたように，Z'sSTAFFは1番目のパラメータとしてダミーのテキストVRAMのアドレスを渡すのに対して，Z's-EXは必ずメインメモリ内のアドレスを渡します。それを利用して，テキストVRAMのアドレス($E00000)が渡されたらZ'sSTAFFから起動されたと判断できます。

登録方法

リスト2を6月号の付録ディスクについてきたMAC.Xで打ち込み，LHAで展開することで実行ファイルが作成されます。キー操作は，

```
A>MAC↵
Y            (新規作成の場合のみ)
LR.LZH↵      (ファイル指定)
C            (CRCモード)
E            (エディットモードへ)
(データを打ち込む)
ESC          (コマンドモードへ)
S            (セーブ)
4468↵        (セーブバイト数指定)
```

のようになります。ファイルの展開は，

```
A>LHA E LR
```

です。

このプログラムは画面全体もしくは矩形範囲に対応しています。

呼び出し方法は，

```
LINEREVISE Address
        [X1 Y1 X2 Y2] [P1 [P2]]
```

となります。ここで，パラメータP1はカウントするドットの最大値を決定し，パラメータP2はにじませる濃度を決定します。双方とも値が大きいほど効果が強く，仕様上の設定範囲は，P1は1～9（省略時6），P2は0～9（省略時5）です。が，Z'sSTAFFから無理やり32などの値をぶちこんでみても，変な特殊効果として面白いかもしれません。

また，オプションは渡しても認識はせず，全画面か矩形かはパラメータの数で認識します。

登録は図5のように行ってください。ここで，Z'sSTAFFでの登録は例によって'+'を忘れずにつけてください。これを忘れると，メニューバーやウィンドウが保護色と化して「どこへいったんだ～！」と叫ぶハメになります。ウィンドウが質素だと思われる方はご自由にどうぞ。ただし，ウィンドウの下は書き込まれないでしょう。

図3　2×2ビットパターン

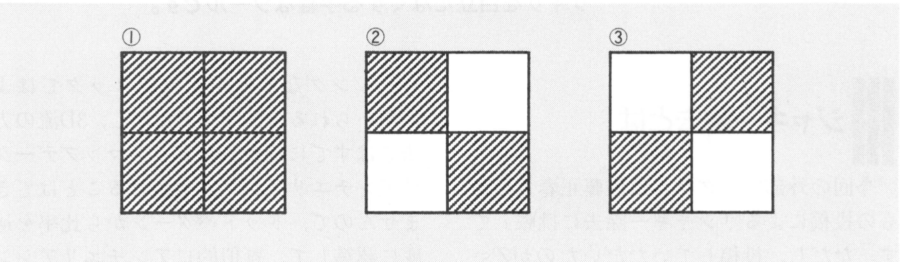


図4　カウント例

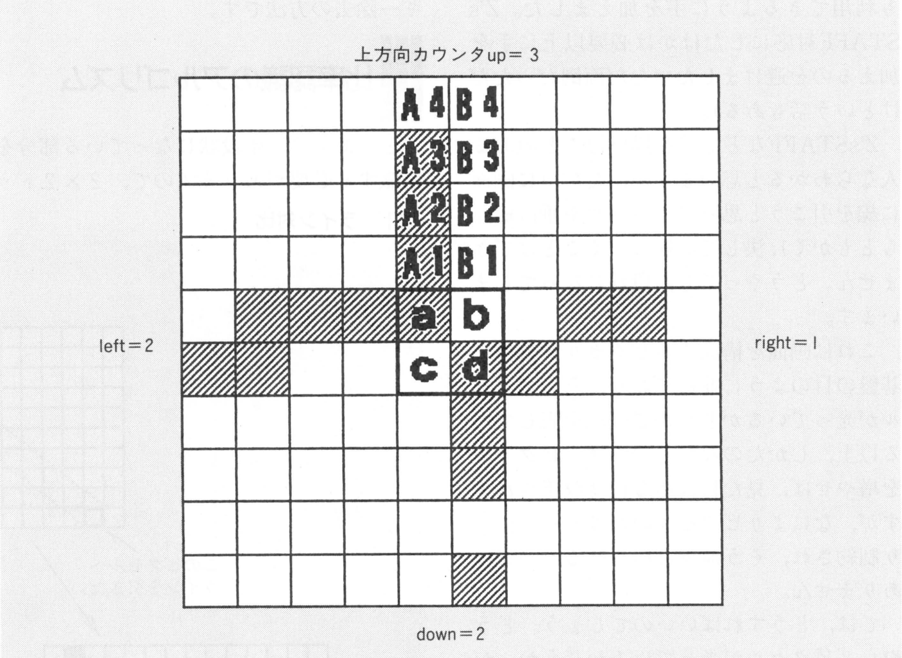


図5　登録方法

Z'sSTAFFでの登録

	パラメータを省略しない	濃度パラメータを省略	両パラメータ省略
画面全体の場合	LINEREVISE %% /A+	LINEREVISE % /A+	LINEREVISE /A+
矩形指定する場合	LINEREVISE %% /B+	LINEREVISE % /B+	LINEREVISE /B+

Z's-EXでの登録（Z's-EX内で）

	パラメータを省略しない	濃度パラメータを省略	両パラメータ省略
画面全体の場合	：ジャギー除去 0,2：1-9,6：0-9,5 LINEREVISE	：ジャギー除去 0,1：1-9,6 LINEREVISE	：ジャギー除去 0,0 LINEREVISE
矩形指定する場合	：ジャギー除去 1,2：1-9,6：0-9,5 LINEREVISE	：ジャギー除去 1,1：1-9,6 LINEREVISE	：ジャギー除去 1,0 LINEREVISE

使い勝手

佐藤さんのお便りの中にもありましたが，ブラシなどのある絵には悪影響があります。また，入り組んだ絵の場合，本来はにじませてほしくない部分までにじんでしまいます。もし，このラインをにじませたくなかったら，ラインをマスクしておけばマスクの下はもちろん書き込みしませんし，見ることもしません。

また，反対にマスクの下も見てほしいというのであれば，71～72行目の‘&&(!(Tram[i+x]&1))’と‘&&(!(Tram[i+x+1]&1))’を取ってコンパイルしてください。

けっこう面倒なことをしている割に，そこそこのスピードで動きます。なかなかよくできたプログラムといえるでしょう。佐藤さんはZ's-EX ver.1.0の頃から自作プログラムを組み込んでいたそうです。これからもがんばってください。なにを隠そう，私もZ's-EXを改良して持ち込んだのがライターになるきっかけでした。皆さんも外部プログラムなど，作ったらなんでもガンガン送ってみてください。

グラフィック共通資産へ向けて

さて，Z'sSTAFFの外部拡張をうけてZ's-EXは裏画面の合成などの特殊機能をさらに生かしたツールへとバージョンアップが進められています。

エフェクト関係もいっそうの拡充が必要でしょう。EPA2やMATIERなどを見るとなかなか面白いエフェクトが揃っていますね。アイデア次第で無限に拡張できるところがZ's-EXシステムの利点ですから，これらのものも徐々に取り込まれていくと思います(?)。

これらの外部ファイルはZ'sSTAFFやZ's-EXだけでなく，フリーソフトウェアのMFGEDなどでも利用できるように対応されており，X68000でのグラフィック共有資産としての性格を持ちつつあります。また，今後はMATIERの子プロセス起動にも対応できるような，コマンドライン起動での動作も考慮した外部ファイル作りがいっそう重要になってくるでしょう。どうせ作るならなんにでも使えたほうがいいでしょうから。

このような傾向が進むとグラフィックツールの機能較差がどんどん小さくなります。Z'sSTAFFにしてもMATIERにしてもMFGEDにしても，ユーザーインタフェイスはまったく違いますから，ユーザーは機能よりも「自分の扱いやすいもの」という基準でツールを選ぶことができるようになるのです。

そのためには対応する外部ファイルが多ければ多いほどよいわけです。現在でもいくつかの意欲的な投稿作品が届いていますが，もっともっと必要です。外部ファイルはそのうちかき集められてMOOK化されると思いますので有志の皆さんの幅広い協力をお願いいたします。

リスト1

```
  1: /**************************************************************
  2: *  Zs_EX (Ver1.10) & Zs-STAFF (Ver3.0) の外部コマンド
  3: *  ジャギー除去に挑戦
  4: *  LINEREVISE.X                               1992.4  佐藤 正春
  5: *
  6: *<Another Address> [ x1, y1, x2, y2 ] [parameter1][parameter2]
  7: *  全画面又は矩形・Zs-STAFF からは'+'オプションをつける
  8: *  parameter1:補色ドット数 1~9     省略時6
  9: *  parameter2:濃度         0~9     省略時5
 10: *
 11: **************************************************************/
 12:
 13: #include<iocslib.h>
 14: #include<stdlib.h>
 15: #include<stdio.h>
 16: #include<doslib.h>
 17:
 18: unsigned short   *Gram= (unsigned short *)0xC00000;
 19: unsigned short   *Tram= (unsigned short *)0xE00000;
 20:
 21: int   x1= 0, y1= 0, x2= 511, y2= 511;
 22: int   parm1= 6, parm2= 5, pm, adj;
 23:
 24: void judge(int,int,int);
 25: void set(int,int,int,int,int,int,int);
 26: void c_set(int,int,int);
 27:
 28: int main(ac,av)
 29: int ac;
 30: char *av[];
 31: {
 32: int   x, y;
 33: int   i, j, ssp;
 34:
 35:    if( ac<2 ){
 36:       B_PRINT((unsigned char *)"LINEREVISE.X¥r¥n¥r¥n");
 37:       B_PRINT((unsigned char *)"Zs_EX & Zs-STAFF専用外部コマンド
EFFECTプログラム¥r¥n");
 38:       B_PRINT((unsigned char *)"コマンドからは、実行出来ません。¥r¥n");
 39:       return( 0 );
 40:    }
 41:    if( ac>=3 && ac<=4 ){
 42:       parm1= atoi( av[2] );
 43:       parm2= atoi( av[3] );
 44:    }
 45:    if( ac>5 ){
 46:       x1= atoi( av[2] );
 47:       y1= atoi( av[3] );
 48:       x2= atoi( av[4] );
 49:       y2= atoi( av[5] );
 50:       if( ac>6 ){
 51:          parm1= atoi( av[6] );
 52:          parm2= atoi( av[7] );
 53:       }
 54:    }
 55:
 56:    pm=parm1*256/9;/* 9 は パラメータ1 max 値 */
 57:    adj=35+parm2*4;/*adj maxは分母(c_set()では128)の50%位。ここでは71*/
 58:    ssp=SUPER( 0 );
 59:
 60:    for( y=y1; y<y2; y++){
 61:       if( MS_GETDT()&0x00ff ){    /*右ボタンで中断*/
 62:          while( MS_GETDT()&0x00ff );
 63:          break;
 64:       }
 65:       i= y*512; j=(y+1)*512;
 66:       for( x=x1; x<x2; x++){
 67:          if( (Tram[i+x] == Tram[j+x+1])&&    /* 4dot 同色 */
 68:              (Tram[i+x+1] == Tram[j+x])&&
 69:              (Tram[i+x] == Tram[j+x]) )continue;
 70:
 71:          if( (Tram[i+x] == Tram[j+x+1])&&(!(Tram[i+x]&1)) ) ju
dge(  x, y , 0 ); /*左上斜め同色 */
 72:          if( (Tram[i+x+1] == Tram[j+x])&&(!(Tram[i+x+1]&1)) )
judge(  x, y , 1 ); /*右上斜め同色 */
 73:       }                       /*    ↑マスクの下は見ない    */
 74:    }
 75:
 76:    if( atoi( av[1] )==0xE00000 ){/* たぶんZs-STAFFからの起動だろう */
 77:       y2 = y;
 78:       for( y=y1; y<=y2; y++ ){/*処理したところまでTramにコピーするのさ*/
 79:          for( x=x1; x<=x2; x++ ){
 80:             if( Tram[x+y*512]&1 ) continue;    /* マスクかな? */
 81:             Tram[x+y*512] = Gram[x+y*512];
 82:          }
 83:       }
 84:    }
 85:
 86:    SUPER( ssp );
 87:    return( 0 );
 88: }
 89:
 90: void judge( x, y ,sw )
 91: int   x, y, sw;
 92: {
 93: int   up=0, down=0, left=0, right=0;
 94: int   i, j, k, m, mm, max= 15, xy;
 95:
 96:    xy= y*512+x;
 97:    j= (sw)? (xy+1): xy;
 98:    k= (sw)? xy: (xy+1);
 99:    m= 0;
100:    if(Tram[j] == Tram[k]) goto set0;
101:    for( i=(y-1); i>=y1; i-- ){          /*  上に伸びているか。  */
102:       m++; mm= m*512;
103:       if( (Tram[j] == Tram[j-mm] )&&(Tram[j] != Tram[k-mm]) ){
104:          up+=1; if( up>max ) break;
105:       } else  break;
106:    }
107: set0:
108:    j= (sw)? (xy+512): xy;
109:    k= (sw)? xy: (xy+512);
110:    m= 0;
111:    if(Tram[j] == Tram[k]) goto set1;
112:    for ( i=(x-1); i>=x1; i--){          /*  左に伸びているか。  */
113:       m++;
114:       if( (Tram[j] == Tram[j-m] )&&(Tram[j] != Tram[k-m]) ){
115:          left+=1; if( left>max ) break;
116:       } else  break;
117:    }
118: set1:
119:    j= (sw)? (xy+512): (xy+512+1);
120:    k= (sw)? (xy+512+1): (xy+512);
121:    m= 0;
```

```
    if( Tram[j] == Tram[k] ) goto set2;
    for ( i=(y+2); i<=y2; i++){        /*  下に伸びているか。  */
       m++; mm= m*512;
       if( (Tram[j] == Tram[j+mm] )&&(Tram[j] != Tram[k+mm]) ){
          down+=1; if( down>max ) break;
       } else  break;
    }
set2:
    j= (sw)? (xy+1): (xy+512+1);
    k= (sw)? (xy+512+1): (xy+1);
    m= 0;
    if( Tram[j] == Tram[k] )goto set3;
    for ( i=(x+2); i<=x2; i++){        /*  右に伸びているか。  */
       m++;
       if( (Tram[j] == Tram[j+m])&&(Tram[j] != Tram[k+m]) ){
          right+=1; if( right>max ) break;
       } else  break;
    }
set3:
    set( x, y, up, left, down, right, sw );
}

void set( x, y, up, left, down, right, sw )
int   x, y, up, left, down, right, sw;
{
int   f0, f1, i, j, k, m;
int   L1, xy;

/* up:上 ; left:左 ; down:下 ; right:右
    k: 参照色no;        j:GRAM 書き込み位置;
    xy :4ドット角左上のバッファ位置
*/
    xy= y*512+x;
    j= (sw)? xy: (xy+1);
    k= (sw)? (xy+1): xy;
/*上に伸びているドットが、0個でGRAM 書き込み位置がすでに処理されているときskip1 へ*/

    if( up==0 ){
       if( Tram[j] != Gram[j] ) goto skip1;
    }

/*上に伸びているドットが、0個でなく、GRAM 書き込み位置がすでに処理されているとき、
up 変数に0を代入する。 */

    if( (j-512) >= 0){
       if( Tram[j-512] != Gram[j-512] ) up=0;
    }
    if( up>0 ){
       L1= (up*pm+200)/256;    /*上*/
       if( !L1 ) L1=1;
       f0= adj/(L1+1);
       if((y-L1+1)<y1) L1= y-y1+1;
       for ( i= 1; i<=L1; i++ ){
          f1= adj-f0*i;
          c_set( k, j, f1);
          j-= 512;
       }
    } else {
       c_set( k, j, adj/4);
    }

skip1:
    j= (sw)? (xy+512+1): (xy+512);
    k= (sw)? (xy+512): (xy+512+1);
    if( !down ){
       if( Tram[j] != Gram[j] ) goto skip2;
    }
    if( (j+512) < 262144 ){
       if( Tram[j+512] != Gram[j+512] ) down= 0;
    }
    if( down>0 ){
       L1= (down*pm+200)/256;    /*下*/
       if( !L1 ) L1=1;
       f0= adj/( L1+1 );
       if( (y+L1) > y2 ) L1= y2-y;
       for ( i= 1; i<=L1; i++ ){
          f1= adj-f0*i;
          c_set( k, j, f1);
          j+= 512;
       }
    } else {
       c_set( k, j, adj/4);
    }

skip2:
    j= (sw)? xy: (xy+512);
    k= (sw)? (xy+512): xy;
    if( !left ){
       if( Tram[j] != Gram[j] ) goto skip3;
    }
    if( x > 0 ){
       if( Tram[j-1] != Gram[j-1] ) left= 0;
    }
    if( left>0 ){
       L1= (left*pm+200)/256;    /*左*/
       if( !L1 ) L1=1;
       f0= adj/(L1+1);
       if( (x-L1+1)<x1 ) L1= x-x1+1;
       for ( i= 1; i<=L1; i++ ){
          f1= adj-f0*i;
          c_set( k, j, f1);
          j--;
       }
    } else {
       c_set( k, j, adj/4);
    }

skip3:
    j= (sw)? (xy+512+1):(xy+1);
    k= (sw)? (xy+1): (xy+512+1);
    if( !right ){
       if( Tram[j] != Gram[j] ) goto skip4;
    }
    if( x < 510 ){
       if( Tram[j+1] != Gram[j+1] ) right= 0;
    }
    if( right>0 ){
       L1= (right*pm+200)/256;    /*右*/
       if( !L1 ) L1=1;
       f0= adj/(L1+1);
       if( (x+L1)>x2 ) L1= x2-x;
       for ( i= 1; i<=L1; i++ ){
          f1= adj-f0*i;
          c_set( k, j, f1);
          j++;
       }
    } else {
       c_set( k, j, adj/4 );
    }
skip4:
    return;
}

void c_set( no1, no2, f )
int  no1, no2, f;/* no1 参照位置;no2 書き込み位置;f 参照色割合(/128); */
{
unsigned int   r0, g0, b0;
unsigned int   r1, g1, b1;
unsigned short   col;

    if( Tram[no1] == Tram[no2] ) return;
    if( Tram[no2] & 1 ) return;  /* マスクだよ〜ん */
    if( f>=128 ) {
       Gram[no2]= Tram[no1];
       return;
    }
    if( f<=0 ) return;
    col= Tram[no1];                        /*参照位置の色*/
    g0 = (col>>3)&0x1F00;                  /* 256倍のゲタを履いています*/
    r0 = (col<<2)&0x1F00;
    b0 = (col<<7)&0x1F00;
    col = Tram[no2];                       /*書き込み位置の色*/
    g1 = (col>>3)&0x1F00;
    r1 = (col<<2)&0x1F00;
    b1 = (col<<7)&0x1F00;

    g0 = ( (g0*f)+(g1*(128-f))+64 ) >>7; /* 128で割って四捨五入 */
    r0 = ( (r0*f)+(r1*(128-f))+64 ) >>7;
    b0 = ( (b0*f)+(b1*(128-f))+64 ) >>7;

    g0 = (g0<<3)&0xF800;
    r0 = (r0>>2)&0x07C0;
    b0 = (b0>>7)&0x003E;

    Gram[no2]= ( (unsigned short)r0 | (unsigned short)g0 | (uns
igned short)b0 );
}
```

リスト2

```
0000   25 72 2D 6C 68 35 2D 4C   : 46
0008   11 00 00 2C 1E 00 00 DC   : 37
0010   70 CE 18 20 01 0C 6C 69   : 58
0018   6E 65 72 76 69 73 65 61   : 61
0020   2E 78 0F D6 48 00 00 0E   : E1
0028   CB 7C DE FB D6 34 A3 BE   : 8B
0030   FE F7 A6 CD 9E 9A EC 8F   : 1B
0038   88 EB 3B 48 6D 1F 1E 9A   : 3A
0040   C6 59 42 6E 86 18 58 19   : DE
0048   00 DD F1 07 B5 D6 5D 91   : 4E
0050   D1 A5 D5 C3 17 D0 64 F7   : 50
0058   84 90 93 7A 12 14 1C 28   : 8B
0060   FB 77 60 70 97 16 45 46   : 7A
0068   A3 79 C7 0A E3 97 9E 6E   : 73
0070   64 E9 63 16 F2 E4 C3 24   : 83
0078   31 E7 04 79 B7 51 70 E3   : F0
-------------------------------------
SUM:   E1 A6 AE BE AD 4B 04 6F   FA8E
```

```
0080   8C 78 82 F2 03 4C 28 4E   : 3D
0088   32 61 90 33 6C F7 EF DF   : 87
0090   FE F4 90 45 EC C6 E7 75   : D5
0098   EB 7A B7 E0 EF 92 E5 CB   : 2D
00A0   F3 5E F8 6F F0 AE F7 76   : C3
00A8   91 B7 1B 92 C4 FE 30 6F   : 56
00B0   F7 65 38 03 3D D0 1B 09   : C8
00B8   00 3A E3 00 1D CB 0F DB   : EF
00C0   FA BF BA 8D AD 8F AD D7   : C0
00C8   F7 5A ED 6D 5C 8A 27 54   : 0C
00D0   4E 82 5E E3 6B E0 CB B3   : DA
00D8   97 FC A9 4A F3 12 1B 7A   : 20
00E0   29 77 32 F6 B2 D5 97 69   : 4F
00E8   2E D2 FB 9F E6 C9 DF 58   : 80
00F0   79 9B F9 91 E6 7F 34 CF   : 06
00F8   2E 66 C8 49 44 28 D3 B1   : 95
-------------------------------------
SUM:   F6 DC 23 E4 81 32 6B CF   8011
```

```
0100   97 B5 99 B6 99 E3 CC B0   : 93
0108   99 E3 4C 60 50 59 84 1F   : 74
0110   7A 5F 3A 5F BD 2A 1F 91   : 09
0118   89 E3 ED 6F 6E 3C DD EC   : 3B
0120   BF 8A 5F B1 2F F3 CA 50   : 95
0128   48 82 27 3B B6 7B 51 00   : AE
0130   12 A9 B8 07 FB A3 B8 53   : 23
0138   36 89 C4 F1 19 BA FB 3D   : 7F
0140   50 04 77 B1 BC B1 B7 B8   : 58
0148   35 3E BC 6F 2E 48 58 74   : E0
0150   81 DA A0 7C 2D D1 FC 29   : 9A
0158   96 D5 1B A7 02 5A 94 0D   : 2A
0160   08 20 62 DE 1C 6F 2E 9F   : C0
0168   10 01 F3 81 02 ED BB 68   : 97
0170   58 D8 6E A7 C4 AF BF FD   : 74
0178   04 B9 EC 8C D2 5C F6 48   : A1
-------------------------------------
SUM:   92 BB AB 9D DA F8 57 DA   9820
```

```
0180  DB 45 BB 73 86 DB C6 CF  : 44
0188  49 5E 8E E9 C8 9F D3 08  : 60
0190  57 BD 74 FB EC D4 FA 5A  : 97
0198  72 07 F1 B1 13 6E 14 30  : E0
01A0  02 37 B0 01 9C AC B1 B0  : 93
01A8  30 DB E3 0B 43 98 C6 EA  : 84
01B0  9E 6F DB 30 1D F8 92 98  : 57
01B8  8F B4 D0 D4 42 AA D3 92  : 38
01C0  EE 45 87 B3 EC 12 AE D8  : F1
01C8  03 9E 61 3A 6C 4C 31 71  : 96
01D0  CC AF 49 F3 DD 5B A6 A8  : 3D
01D8  07 17 43 3F BA A1 E9 7F  : 63
01E0  5A 53 F1 D8 E9 70 6A 08  : 41
01E8  F7 23 D7 F6 0B 19 89 DF  : 73
01F0  A3 53 3D C8 61 27 33 66  : 1C
01F8  25 B8 10 38 E9 87 1A 52  : 01
-------------------------------------
SUM:  29 C6 75 05 B8 33 31 34  6D04

0200  4E 26 B1 A5 82 F4 30 A4  : 14
0208  A0 7A D4 70 60 A0 18 12  : 88
0210  5E 9E B3 8C 05 17 B3 CD  : D7
0218  29 9A C2 6E 38 D2 B3 06  : B6
0220  BD 40 55 CB 52 16 27 81  : 2D
0228  3F 02 78 B0 19 F8 E4 DD  : 3B
0230  FE A3 13 C4 C4 3A E6 27  : 83
0238  7D 06 31 EF 23 CC 6B 0F  : 0C
0240  ED 07 20 47 DD B5 99 C6  : 4C
0248  D9 34 D7 04 4E 8F 0E 45  : 18
0250  86 06 18 0C 4F AD E0 53  : DF
0258  FE 12 1A D3 13 68 A6 DE  : FC
0260  2B 79 2C A9 E8 7C F9 BE  : 94
0268  DE A1 F9 30 9F 45 32 07  : C5
0270  39 29 E9 B2 E6 B0 84 10  : 27
0278  04 0E 59 CE 8F FB 50 04  : 17
-------------------------------------
SUM:  7C 67 9B C0 FA 56 36 32  2F24

0280  02 3C CC 19 96 DC DC B5  : 26
0288  9D C7 0E 07 64 D3 75 92  : B7
0290  12 2C 2B F1 9E E4 C6 BB  : 5D
0298  8D AF 67 28 61 ED F7 DC  : EC
02A0  F7 BC DF 77 CE FC 5E 67  : 98
02A8  17 49 B9 0E CD 2A B0 77  : 45
02B0  B9 A1 FF 2E 0D 58 72 AF  : 0D
02B8  C1 08 B7 E0 8A EE 6A 68  : AA
02C0  23 DE 2A 7D 4D 01 AE B8  : 5C
02C8  AA A0 54 7E C7 DD ED 81  : 2E
02D0  AC 36 4D 8D 39 FE 34 AA  : D1
02D8  D1 6C F8 72 50 05 F5 EF  : E0
02E0  C5 07 AE 97 84 58 9F 27  : B3
02E8  69 13 34 97 52 23 59 CA  : DF
02F0  8F 31 39 E1 50 FF 52 4E  : C9
02F8  2F A5 4D 62 B4 5B AE 1D  : 5D
-------------------------------------
SUM:  FC 9C E5 37 A2 A2 B4 01  B2CB

0300  BB 39 4B 24 94 FF CF 7D  : 42
0308  74 63 FF 17 AB EF 6D 1D  : 11
0310  65 A1 ED B8 86 5E BA A9  : F2
0318  A0 E4 0C 06 9E 1D 8C 1E  : FB
0320  85 55 07 24 C1 7F 5E 13  : B6
0328  EE 47 09 A0 DA 2F 23 69  : 73
0330  59 9A 4B A4 82 4C A4 70  : C4
0338  5C C9 AC 60 89 4A CF 87  : 5A
0340  03 B7 6C 70 1D 7F AB F8  : D5
0348  B3 30 1E 45 B7 1F E9 C0  : C5
0350  68 37 64 E6 83 B7 81 4E  : F2
0358  FD B8 3C CC 8F BD 32 5C  : 97
0360  5F 2F FB B2 39 BF 4B 23  : A1
0368  9B F4 32 3E EF D5 F7 76  : 30
0370  8E A3 B6 87 B8 E2 75 DD  : 5A
0378  7F FA E1 BF AD DD 99 93  : CF
-------------------------------------
SUM:  7E B6 38 5E 7C 12 0D 3F  1BC5

0380  B5 85 C6 18 FE A6 4F 37  : 42
0388  AE C9 E6 E6 B2 78 BE AF  : DA
0390  17 2F 03 E7 A7 5C 71 E0  : 84
0398  55 82 30 1D 40 EC A0 3B  : 2B
03A0  81 44 36 DD 8F D4 0A 1F  : 64
03A8  67 EE 46 45 5F 57 F4 6B  : F5
03B0  32 59 E5 46 F0 0D 10 39  : FC
03B8  25 10 29 89 7E 8A DC 1E  : E9
03C0  33 3A AA DC 1E 8D 97 CC  : 01
03C8  E4 94 44 DD 50 F3 38 35  : 49
03D0  30 BF 27 44 D1 12 92 EE  : BD
03D8  34 2E D5 1F B1 F7 86 BA  : 3E
03E0  E7 EC F4 00 E5 4F A8 AF  : 52
03E8  8B 28 70 6B 58 A5 C8 E4  : 37
03F0  F3 49 5E 97 46 81 4F A8  : EF
03F8  2C 01 C7 10 B8 6C B3 94  : 6F
-------------------------------------
SUM:  1A B3 DC 21 1E 92 61 5A  38D6

0400  16 A9 10 F4 A1 7B FD 53  : 2F
0408  4A 9C 2F 83 AA 62 9F DF  : 22
0410  9C 18 5B 29 C0 B9 F7 21  : C9
0418  16 55 C0 06 FA 01 69 67  : FC
0420  7D 00 E3 49 7D 6D 8D 85  : A5
0428  48 13 49 0E 26 CD C9 AB  : 19
0430  67 97 78 B1 FB 73 8D A6  : C8
0438  CE FD FD 74 4D 98 CF AE  : 9E
0440  66 9E B5 60 58 DE FA F2  : 3B
0448  3B 06 99 9C 9C 16 2E 10  : 66
0450  A9 10 3C 09 51 EF 36 C7  : 3B
0458  7E 09 9B 4E E6 AA 81 F9  : 7A
0460  8E 77 FB B8 07 C0 72 D4  : C5
0468  CA BA 43 E2 36 21 2D 0C  : 39
0470  F9 3F 6E 3C CF C1 32 DB  : 7F
0478  92 AB E6 94 36 C7 EB D8  : 77
-------------------------------------
SUM:  B7 31 B2 DF 5D D2 49 93  E595

0480  F9 58 80 19 A5 64 4A 66  : A3
0488  02 EB 42 EF E2 6E 1C 5B  : E5
0490  B2 19 6C 64 62 75 C0 D2  : 04
0498  77 B2 3B 84 FF C3 1D E4  : AB
04A0  93 1D 23 91 D2 16 79 96  : 5B
04A8  6D FF 53 23 70 2C 8D C0  : CB
04B0  B2 3F B9 EF 12 44 59 2E  : 76
04B8  F1 B6 4B 45 FB 99 2C DB  : D2
04C0  C2 64 6F 83 05 8B 67 35  : 44
04C8  27 70 2F C1 DF D6 C8 4B  : 4F
04D0  12 FF E3 21 28 31 FF 46  : B3
04D8  36 50 6B 7E CC EF AE 0D  : E5
04E0  C5 02 85 C3 FF D2 C3 D9  : 7C
04E8  CA C2 D9 0C C0 B9 BA 0D  : B1
04F0  12 8C CE 24 58 B1 3D EB  : C1
04F8  6C 4E AD 99 DF FA C5 50  : EE
-------------------------------------
SUM:  05 E0 A8 47 05 E0 29 CA  812C

0500  CC 83 BC 86 E0 ED 0C C8  : 32
0508  3B 53 F4 B1 9D 76 F3 AE  : E7
0510  C5 79 5F F7 06 FF A2 53  : 8E
0518  4D 06 63 93 72 7E B6 32  : 21
0520  EE FC 78 A9 FF 98 21 6F  : 32
0528  FA 98 E8 1F 85 B2 03 DB  : AE
0530  63 B9 24 8F 16 3B 79 98  : 31
0538  C7 72 7F 73 1D 6A 0C F3  : B1
0540  D3 B3 20 CD 0F D8 C6 63  : 83
0548  FF D3 19 36 92 BC 86 F4  : E9
0550  95 C7 2D 25 6A 05 5E 53  : CE
0558  76 3A 49 A2 DB 0D 9E 9A  : BB
0560  C5 53 8E 4E EC B5 0D 83  : 25
0568  06 40 60 43 5C 0E B9 36  : 42
0570  FF A0 76 CE EB C5 C2 F1  : 46
0578  ED A8 CC 53 D0 D1 CA 90  : AF
-------------------------------------
SUM:  BF 76 54 07 95 CE 9A 4E  E6C7

0580  0A 81 DA F6 A4 B4 E8 D9  : 74
0588  A7 E6 3B 3B 9B 13 13 73  : 37
0590  DB 03 4C 7F 53 09 9E B1  : 54
0598  42 2E A6 1B 39 4F C6 0E  : 8D
05A0  F9 92 CA E9 E2 D8 FB 60  : 53
05A8  66 71 88 F2 04 5F 50 45  : 49
05B0  E4 74 88 5E 43 62 F2 2A  : FF
05B8  1C 71 E6 69 20 D6 66 23  : 5B
05C0  F7 41 AA 0F 10 35 07 73  : B0
05C8  37 7C 5C 32 21 DB 85 61  : 23
05D0  DD 2B CE 21 AD 97 1F BE  : 18
05D8  D2 B4 B5 8F 16 FF E5 8B  : 4F
05E0  7B 84 FE B4 71 C3 1B FF  : FF
05E8  1C 52 38 E2 CD 68 08 2A  : EF
05F0  50 B5 98 FC 48 5A 2E E9  : 52
05F8  9C 97 F4 3D 1D F1 61 D3  : A6
-------------------------------------
SUM:  8D 3E 12 2D AB AA 44 FF  EC70

0600  18 CE 68 C4 6C E3 C0 9F  : C0
0608  EC 13 CE 6C 9E 71 89 F2  : C3
0610  3A E4 EE 3B 04 F9 03 8F  : D6
0618  72 79 C6 48 96 3D C9 01  : 96
0620  50 8A 5F E5 DF 65 E6 BA  : 02
0628  0C 42 45 EE 68 FA 1E A1  : A2
0630  66 1B AD 57 E1 BF 67 21  : AD
0638  A6 97 06 AD 9C A6 2D D9  : 38
0640  EB 0F DE A6 8D 8A C0 33  : 88
0648  A9 9A 4E A7 58 95 B7 87  : 63
0650  B1 B9 B2 B9 DC 24 B5 C5  : 4F
0658  CA 52 34 D5 90 60 C1 4A  : 20
0660  46 B3 D9 FC BC 5C 6C 1F  : 71
0668  C6 96 DB 49 57 06 08 6B  : 50
0670  2E 24 EE 2E 7C 35 F6 57  : 6C
0678  49 68 F4 F5 9F C3 A1 AC  : 49
-------------------------------------
SUM:  AA 45 E9 CD E7 4B A5 CC  B20A

0680  AC D0 D6 69 F4 3A 78 29  : 8A
0688  6B A2 43 8F 63 02 25 DD  : 46
0690  9E D6 E2 DC 02 C0 58 ED  : 39
0698  47 2A 9F 8A 3D 0C BA 94  : 31
06A0  FE E8 D3 CF B2 03 FD 3E  : 78
06A8  EF E6 38 29 FD E1 BB EC  : BB
06B0  A5 52 1A 52 E5 7E 97 08  : 65
06B8  D4 F0 C6 FC 13 59 F6 CD  : B5
06C0  4F 16 9F DC 1A 7F BF BB  : F3
06C8  F9 81 14 E8 63 CA E1 E1  : 65
06D0  80 EB 9D C6 16 EE E0 29  : DB
06D8  F1 44 4F 18 A1 E5 1D 20  : 5F
06E0  EA C6 08 85 9B 94 AB 4A  : 61
06E8  B8 EC 62 34 4E C1 88 F8  : C9
06F0  E6 A7 A6 59 CD 2E 7A 48  : 49
06F8  F7 FF 40 DE 8B 41 E4 6F  : 33
-------------------------------------
SUM:  9A A0 74 36 B2 A3 22 64  86B3

0700  54 62 3B 11 5E AE 81 52  : E1
0708  B7 59 5E 9F D2 85 E4 99  : E1
0710  53 A6 25 4A 4D AA 52 02  : B3
0718  D7 84 28 BB 04 FC A3 53  : 34
0720  F2 1B 2E FA FE 57 8F 00  : 19
0728  E8 14 4C C5 B1 BD 8A C5  : CA
0730  A8 1D FD 35 64 AB E3 99  : 82
0738  63 C9 A7 E2 0D 3C DD 60  : 3B
0740  E5 95 38 09 7B 83 F7 0E  : BE
0748  26 B6 1D 8B 58 34 B4 19  : DD
0750  4B A3 23 7B 49 C8 5D CD  : C7
0758  27 21 69 CA F5 0C 1F A2  : 3D
0760  60 FD 38 A8 AE 82 7C 41  : 2A
0768  C4 B3 95 C8 53 C9 50 81  : C1
0770  E1 28 33 2E 9C B8 62 DB  : FB
0778  74 07 EB 0B A4 6A 2A 4A  : F3
-------------------------------------
SUM:  10 E8 D0 0D F3 CC B2 7B  B5A0

0780  80 73 35 D2 2F 51 C5 47  : 86
0788  12 0F E4 8F 0F 07 72 13  : 2F
0790  EB 74 0B 1B B8 EC D2 DC  : D7
0798  8C E9 FA 4A 3D EF 38 8C  : A9
07A0  CE 23 37 DE 97 D0 66 32  : 05
07A8  F3 40 BA 02 56 70 20 AE  : 83
07B0  98 6B 3C BA A6 E2 42 80  : 43
07B8  84 02 3F 36 AB 2F 35 F2  : FC
07C0  07 2A AE 01 ED 25 E0 E8  : BA
07C8  21 0A DE 10 21 A0 DC 90  : 46
07D0  D0 46 BF 18 F9 E6 3B E5  : EC
07D8  3D 30 70 21 5D E7 21 83  : E6
07E0  C8 7F 30 53 37 80 FC F3  : 70
07E8  69 4B 72 64 1A 46 E9 77  : 4A
07F0  C1 24 A8 1B 80 BB CE E6  : 97
07F8  45 EC 41 CA 34 7B 7D 5D  : C5
-------------------------------------
SUM:  52 33 D0 7C DA 12 86 A1  2313

0800  6B 3F 9D 9E DF 60 D5 D9  : D2
0808  23 77 A6 57 4B 4E B4 BA  : 9E
0810  66 A7 28 D4 F5 88 D4 36  : 90
0818  E5 A3 4B 5A F8 F6 B6 E9  : BA
0820  8A 36 7A 72 AE 4A B3 C1  : 18
0828  9E F3 03 D8 27 16 BB 8D  : F1
0830  59 CC D7 B5 C1 F7 B5 F8  : 16
0838  8D 4F 20 93 68 F7 CA B9  : 71
0840  35 7A 86 68 64 5E 9B 04  : FE
0848  F8 AD 8D E7 07 2F 5A 2C  : D5
0850  75 3D E1 31 D7 FB C4 F8  : 52
0858  A2 05 81 C9 2C 3C 17 26  : 96
0860  97 83 15 31 B7 ED 15 B5  : CE
0868  18 33 9F EA 52 3D 16 1F  : 98
0870  80 34 C8 76 70 50 AF D4  : 35
0878  1A 52 5F 1C 76 77 E2 E0  : 96
-------------------------------------
SUM:  74 E9 7A AB 72 2F 8C 87  B683

0880  D6 87 70 BB C1 A5 35 9F  : C2
0888  08 FA C2 B2 BE 91 E6 2A  : D5
0890  BD 31 BB 54 61 AF CA EB  : C2
0898  CC 5A 44 57 A6 30 D7 86  : F4
08A0  0B 47 1F 7A 64 5B 92 45  : 81
08A8  AD CB 52 9A 27 3A 17 05  : E1
08B0  FF 86 CC EA 42 25 D1 2E  : A1
08B8  E4 69 80 B4 12 78 A8 F8  : AB
08C0  96 51 97 7C 6B 32 AC AB  : EE
08C8  96 7D 64 A9 44 5A 3A 6A  : 62
08D0  FC 69 F3 DF D9 6B 98 56  : 69
08D8  8E CA 31 80 E9 15 E8 10  : FF
08E0  38 A7 2C 0C F7 AF 31 61  : 4F
08E8  9B 2E C7 60 D2 E3 D6 34  : AF
08F0  4B E5 57 E3 50 04 6B B1  : DA
08F8  CD E9 0E 1F 78 F2 99 74  : 5A
-------------------------------------
SUM:  A3 B1 65 BC 67 DB 4F DF  F270

0900  2F 2B 41 2C 13 D7 C7 8A  : 02
0908  75 7E 30 C0 0D DC 8B D8  : 2F
0910  D7 6A 4A 52 5B CA 6D FD  : 6C
0918  68 41 C2 AD 07 56 74 7C  : 65
0920  69 55 07 71 A5 0F F9 9C  : 7F
0928  69 6A F3 7E 11 0E 46 A2  : 4B
0930  2E D5 67 1F 1C AE 64 A4  : 5B
0938  25 0F 6A 3B B4 0E 40 FA  : D5
0940  10 A1 04 06 15 51 4C 7E  : EB
0948  D4 45 D7 77 B9 9E 02 D6  : 96
0950  59 95 24 61 DC 0B EF 91  : DA
0958  CC FE 7A FF 1B EF CB F5  : 0D
0960  3C CD AF 93 FD 92 FD 69  : 40
0968  7E D4 BF 44 7B EF FF 74  : 32
0970  AE E8 BF 97 80 12 44 CF  : 91
0978  02 66 C6 64 6F D6 41 B5  : CD
-------------------------------------
SUM:  7B 5F B4 E3 34 FE 9F F2  6546

0980  BA B2 F0 AD 75 01 B2 B4  : E5
0988  B2 93 A8 17 8F 70 0E 24  : 35
```

```
0990  E8 23 58 46 BB 48 6C D9  : F1
0998  55 C5 96 BB 9E E6 14 48  : 4B
09A0  51 AE D3 08 98 B8 2C 2C  : 82
09A8  78 89 17 FF E2 B2 EF 38  : D2
09B0  1A 25 86 78 D9 65 81 C5  : C1
09B8  F1 BE D1 17 A7 11 46 14  : A9
09C0  94 2C 36 CC CF 54 9E DD  : 60
09C8  0D A1 39 5A 2F 0B 9E 34  : 4D
09D0  FF 02 EF 86 9D 12 CF 75  : 69
09D8  9F 33 2C D9 23 16 5A CE  : 38
09E0  2B 98 24 7B 00 59 CE 91  : 1A
09E8  E8 8D 83 69 52 34 EA 2C  : FD
09F0  D3 3E 18 76 78 18 79 DF  : 87
09F8  3B CE F3 B6 2E E6 B0 FC  : 72
-----------------------------------
SUM:  DD 7A 03 F0 0D 91 68 22  A35B

0A00  39 9B D1 EC CF 2A CF 92  : EB
0A08  95 6F 8A CB 2C DD 32 2D  : C1
0A10  66 58 E1 58 00 E1 23 A2  : 9D
0A18  42 62 E0 10 59 D4 BE 90  : 0F
0A20  3A 92 3B 48 8A 4C ED 9B  : AD
0A28  CA E6 04 6A 0B BE D9 B8  : 78
0A30  AE 94 AE CA C9 62 55 98  : D2
0A38  0B AD 9E 3D 74 AB 2D 79  : 58
0A40  5F 92 2B 79 22 5F 72 0C  : 94
0A48  62 C6 BB 7E 0A 29 11 4B  : F0
0A50  CC 54 94 D2 85 E6 2C 14  : 31
0A58  22 42 82 39 BE 24 28 90  : B9
0A60  D8 C5 2F 00 4B 2C FE C5  : 06
0A68  89 93 77 F8 2C 2F 7F B1  : 16
0A70  F4 F0 70 C9 CD DA 5F 8A  : AD
0A78  1E 1A FC 28 7D 60 AF 4E  : 36
-----------------------------------
SUM:  55 CD B5 C3 56 FA 8C 9E  7979

0A80  13 F8 78 38 64 ED 2C E8  : 20
0A88  DF 9D 11 47 90 52 0F 97  : 5C
0A90  EF 7C BE 9A A8 95 DE 5C  : 3A
0A98  52 BF 62 A1 01 CB 55 BF  : F4
0AA0  04 DD 7B 58 B1 21 F0 F1  : 67
0AA8  22 43 52 1F C3 2B 0B C6  : 95
0AB0  F0 8F D7 0E 33 03 8F 3E  : 67
0AB8  78 7F 38 6F F4 FF D3 F9  : 5D
0AC0  CF FC FE A0 FF EF C8 7F  : 9E
0AC8  FF E8 02 E0 59 48 A5 6E  : 7D
0AD0  4D D1 45 4D 7F 48 74 C2  : AD
0AD8  27 27 0E 59 39 37 38 82  : DF
0AE0  E4 DA 7D 29 6B 1F 86 26  : 9A
0AE8  2D 41 F1 35 9E DE 2C 48  : 84
0AF0  53 75 82 D2 F8 73 61 C4  : AC
0AF8  85 50 80 BD 51 D2 35 E5  : 4F
-----------------------------------
SUM:  EC BA 48 C1 9A E5 2C D0  E101

0B00  29 69 8B 4A F4 B5 A5 58  : 0D
0B08  DB 57 BC CE A5 09 F8 EB  : 4D
0B10  42 8D 32 7D FD 27 20 BF  : 81
0B18  A9 9D B2 AB 29 F5 46 AD  : B4
0B20  38 E5 B2 0A 35 43 B2 DB  : DE
0B28  D6 85 D9 3C CB 37 A3 44  : 59
0B30  C4 75 4A 67 0A 25 D4 60  : 4D
0B38  D1 97 A2 59 C5 D1 CD C6  : 8C
0B40  AD 41 A3 55 22 8D 58 39  : 26
0B48  60 E9 0C 0D D9 03 C4 9C  : 9E
0B50  0E A6 C3 8A F7 12 75 1A  : 99
0B58  31 54 5C FA E6 35 DB D8  : A9
0B60  D7 7D EF 7E 03 3E DA 2D  : 09
0B68  39 6C B3 08 24 F8 B1 AC  : D9
0B70  38 73 A6 71 FE 3F 38 74  : AB
0B78  95 77 36 55 A2 17 C5 DD  : F2
-----------------------------------
SUM:  BB 57 EE 78 2D AD ED E5  1CE4

0B80  EF 83 66 7C 96 6B F1 80  : C6
0B88  A6 66 03 5E 1D 9A 58 C5  : 41
0B90  32 31 BE D3 F1 BE 3C BF  : 9E
0B98  78 F2 5D FC 2C 0C 3F 96  : D0
0BA0  AF FC D7 75 41 94 0E 17  : F1
0BA8  A9 4E A8 62 B7 62 91 D7  : 82
0BB0  DE A0 38 40 66 03 02 5D  : BE
0BB8  9E CD FC 3C 0F 1A D3 95  : 34
0BC0  25 94 E1 53 F3 CB 46 74  : 65
0BC8  B9 AA CD 31 3D 52 50 DD  : 1D
0BD0  D4 91 BA CE D3 A9 16 C7  : 46
0BD8  8E A7 A8 3B 7A BC ED E2  : 1D
0BE0  0E D9 F1 44 9D 0B ED F0  : A1
0BE8  9F A8 FE 22 BC 2F 27 88  : 01
0BF0  5C 01 52 F9 0C 06 0A B2  : 76
0BF8  1B BA 2A 5B 5A F6 94 F6  : 34
-----------------------------------
SUM:  77 75 B2 43 79 9A 83 94  44ED

0C00  95 CA 4A 4A A8 1C 47 60  : 5E
0C08  AF A5 0B 2D 0A FE 22 50  : 06
0C10  D2 3E 6E FB B7 84 0E 08  : CA
0C18  1A 68 2E E0 83 FA B3 86  : 46
0C20  7C 19 5D C9 B8 90 F3 75  : 6B
0C28  BC F1 F1 86 8B DF 50 64  : 42
0C30  26 01 CD 6F 4B 73 DC 45  : 42
0C38  F4 C1 DA F9 FF 7C 79 78  : F4
```

```
0C40  85 37 79 EE 33 D3 A7 2B  : FB
0C48  2F 44 5F 00 9D 9B EF 3E  : 37
0C50  4B 2A B8 93 DC 5F 43 69  : A7
0C58  93 9B 6B BD E6 6E DC B4  : 3A
0C60  B1 E3 75 4B 6C D5 18 2D  : DA
0C68  B1 5F 5F 27 A0 5E 8D 28  : 49
0C70  6A 79 31 39 CC 42 8D 7C  : 64
0C78  3B D4 9D 1D 82 E1 AA 09  : DF
-----------------------------------
SUM:  1B B0 83 0F 65 87 53 34  4DE9

0C80  5A 7C A2 D9 4B 9F 6B B6  : 5C
0C88  AF D9 D3 16 81 FC C6 0D  : C1
0C90  C0 85 1D 04 A1 F0 54 92  : DD
0C98  BD D9 61 D9 23 A1 6E C2  : C4
0CA0  F5 9A F8 9C E9 E3 D4 DE  : A1
0CA8  6E BB 8D 09 B1 5F 7D 7A  : C6
0CB0  5D 9F D7 75 57 E4 19 73  : 0F
0CB8  E8 47 DE 7E 56 B1 AF F8  : 39
0CC0  33 F3 24 E6 45 ED 7E 34  : 14
0CC8  2C 48 00 CD 10 E2 91 D1  : 95
0CD0  54 9E 09 9D BF 4D 25 12  : DB
0CD8  A6 5D E2 70 91 7A BD 25  : 42
0CE0  26 75 7C 7C 03 96 E8 60  : 74
0CE8  56 86 A5 6E 89 2A D1 CE  : 41
0CF0  AF CA 35 DE 42 54 31 AE  : 01
0CF8  F2 A8 8B 69 7A 03 F5 43  : 43
-----------------------------------
SUM:  A4 91 1D 55 C4 B0 DC 35  1ED9

0D00  B9 19 2D 7F 7A A1 4D DA  : C0
0D08  9B C6 D7 2E DB 5C F5 23  : B5
0D10  EF 37 CC 66 2E F9 A9 39  : 61
0D18  63 76 65 EF 2D A8 47 59  : A2
0D20  D3 4C 7C 93 AA 30 FC 75  : 79
0D28  DD 7F 54 27 B1 07 69 1E  : 16
0D30  D1 78 23 02 A8 F9 42 F4  : 45
0D38  D7 A8 89 CE 55 21 E3 82  : B1
0D40  30 4A 8D 50 3C 6B 82 FB  : 7B
0D48  DA 53 74 D5 C6 70 E6 FA  : 8C
0D50  52 EE 10 E1 83 91 82 F7  : BE
0D58  1E D6 1B 3F 99 96 7B 75  : 6D
0D60  07 B0 4B 0B 9E 4C 74 92  : FD
0D68  A9 12 54 0E 40 5B 5A 43  : 55
0D70  C5 4E 79 64 A0 3A 69 8C  : BF
0D78  8E 0D FC 6E 0E 10 63 0E  : 94
-----------------------------------
SUM:  7B F5 F1 BC B2 E2 BB 68  46C9

0D80  A4 9F 20 B9 DD B3 41 54  : 41
0D88  63 35 5B 9D 4F 30 A7 B5  : 6B
0D90  5D 11 47 D2 AE E9 B1 37  : 06
0D98  ED 09 30 3A 31 C0 74 AD  : 72
0DA0  87 24 2E 33 83 C4 08 C2  : 1D
0DA8  5D F6 19 39 76 27 05 33  : 7A
0DB0  5E 5E D3 3D 99 FB 39 25  : BE
0DB8  01 6F 5D A6 1F 99 8F 83  : 3D
0DC0  11 28 BF 57 1F 5A EF F6  : AD
0DC8  E3 EA 3C 3E 58 5E 98 B8  : 4D
0DD0  F9 B7 C7 87 B1 C7 DD 57  : AA
0DD8  46 E3 FA 4A 74 18 EF 32  : 1A
0DE0  F8 FF E5 C1 9E 6E FF 63  : 0B
0DE8  A6 CD 00 B7 78 62 D8 C3  : 9F
0DF0  56 64 DE 7B 95 16 10 F1  : BF
0DF8  3F F9 05 96 B0 38 2C F5  : DC
-----------------------------------
SUM:  FA AA ED A0 B3 C0 48 CD  06A5

0E00  DF 0D 2E 97 18 6F 45 13  : 90
0E08  0F DF AA EB 32 DF A3 5F  : 96
0E10  8C 71 31 57 43 DC DD 88  : 09
0E18  90 8D 8E DC 8D 33 EB 49  : 7B
0E20  8B 2E E2 D9 38 A3 4E DD  : 7A
0E28  49 48 D3 6B A5 8C 1D BA  : D7
0E30  5D 38 F2 86 B1 E1 8F 16  : 44
0E38  5D BF 3E E9 95 69 2E 99  : 08
0E40  82 8B 59 C4 69 39 73 F5  : 34
0E48  D2 A2 CA 5D E4 08 60 E6  : CD
0E50  07 74 07 B3 08 79 D8 D7  : 65
0E58  7C 18 98 12 AD 3B 98 B2  : 70
0E60  ED 6A EE 4F 30 91 CC F9  : 1A
0E68  D4 9B 1B 50 22 EA A3 F8  : 81
0E70  CA EF C9 57 6A 91 81 D2  : 27
0E78  09 8C BB 47 83 04 2E 92  : DE
-----------------------------------
SUM:  03 90 CB 8B 7E DB 39 42  6BCE

0E80  A1 BE FB E2 F5 EE F0 07  : 16
0E88  C5 02 CF F8 8E 8B 68 A1  : B0
0E90  8A 3E 60 79 4A FC 92 7A  : F3
0E98  C4 3C CE AF D9 26 27 44  : E7
0EA0  E8 A2 4F 57 F3 B0 5A A5  : D2
0EA8  2F 99 D5 41 FB 18 03 CC  : C0
0EB0  BE CD EE 90 26 C2 27 5F  : 77
0EB8  9D B6 D2 E8 C5 B9 FF 23  : AD
0EC0  35 C6 E7 A7 2E 45 E9 35  : 1A
0EC8  59 17 B2 7A D9 A8 BC D6  : AF
0ED0  BE 49 35 EA 9F C2 D6 F3  : 50
0ED8  D9 C8 F5 BA A5 D1 C6 D3  : 5F
0EE0  3E 6A FE 4B 3A 5E D7 72  : D2
0EE8  5A 77 92 4D 2E 6F AD EF  : E9
```

```
0EF0  D9 9E 9E FB 2D CF 8E AA  : 44
0EF8  FE 0B 30 DA 47 C6 E3 45  : 48
-----------------------------------
SUM:  BA 70 FD 44 A6 C0 CA 7A  8F5E

0F00  62 DA B8 DC E8 7C FF 8F  : C2
0F08  80 2C F6 52 FA 81 A6 5A  : 6F
0F10  FA 5E 5F 02 B9 87 65 C3  : 21
0F18  62 C5 7D 09 AB 4E DE 9A  : 1E
0F20  02 05 A1 3D 35 1A 30 41  : A5
0F28  94 BC 13 C8 2B 26 3B E0  : 97
0F30  49 66 29 27 5D AB 8F 27  : BD
0F38  38 AF 40 0F 3D EC FD 08  : 64
0F40  3B 18 1E CE 52 D2 D2 A3  : D8
0F48  44 7D 15 15 16 71 F3 EA  : 4F
0F50  52 FC 32 35 44 88 C0 CB  : 0C
0F58  92 81 91 98 D3 C5 EE FA  : BC
0F60  05 C8 9C 6B DF 0D 67 2D  : 54
0F68  B0 1A 50 5B 6C 34 BB 5A  : 2A
0F70  C4 69 A3 58 7D 36 E1 CA  : 86
0F78  2B 23 07 7C 8B 21 B7 6B  : 9F
-----------------------------------
SUM:  5C 7F 33 BE 12 D1 0C A4  6071

0F80  4A 34 F5 84 0D 10 9A E2  : 90
0F88  0D B9 B6 A0 1C 25 6A 45  : 0C
0F90  88 BB B9 48 D2 02 0B 38  : 5B
0F98  17 1A 02 2F DB 11 7E 8C  : 58
0FA0  02 B3 35 5B 31 A7 D2 B5  : A4
0FA8  C2 35 DE 1A 07 05 02 D1  : CE
0FB0  C9 4E BD 7A 5F 60 40 E1  : 2E
0FB8  AB 5C 34 EF 55 E5 59 78  : 35
0FC0  96 B7 C6 23 F4 15 D2 59  : 6A
0FC8  6D AD 6F 0D 6F CA 31 7A  : 7A
0FD0  62 5F D9 2D 4A 34 BA 59  : 58
0FD8  02 D5 2D 39 63 2A 24 7B  : 69
0FE0  7C AB 93 57 FD 06 21 7A  : AF
0FE8  62 1A 46 C8 5C 8E 88 AE  : AA
0FF0  95 2D 12 FB 23 11 DF A5  : 87
0FF8  DB 23 88 8E 2C 11 E4 07  : 3C
-----------------------------------
SUM:  E3 01 18 B7 7A 2C 47 45  804A

1000  27 9E A9 F2 72 33 5E FA  : 5D
1008  8C 2A E6 48 F9 67 64 9F  : 47
1010  AB 08 0E 99 D4 EC 2F 49  : 92
1018  AD EA CE 1B 75 AD DB 06  : 83
1020  75 5D ED E9 56 A2 E5 3A  : BF
1028  A9 8F 20 55 3E E9 70 11  : 55
1030  CF 4B 11 70 8D 0F 77 7A  : 28
1038  B1 9D D5 3E E8 DF 52 C4  : 3E
1040  B9 E3 AE 95 82 8B 90 38  : B4
1048  15 A5 B4 79 07 92 74 08  : FC
1050  43 1E 31 06 3B 75 6E D1  : 87
1058  0A 65 91 D5 EE 47 41 11  : 5C
1060  90 32 21 10 AA 16 A0 F1  : 44
1068  AE DA D8 7F 40 7F BF 88  : E5
1070  92 23 C6 FC 5F 88 76 3D  : 11
1078  82 A7 32 D7 D1 7A 63 33  : 13
-----------------------------------
SUM:  16 6F 73 25 89 1C D5 7C  125A

1080  A1 7C 03 E1 FD 17 FB 48  : 58
1088  6A 4B FD 28 FD 40 28 C7  : 06
1090  EB 91 B2 E8 0B D2 C8 E8  : A3
1098  6F E7 C6 FD 20 55 03 F1  : 82
10A0  B6 5F 5D 8F D4 44 1B 93  : C7
10A8  14 97 FA 30 4C 4A 6C 44  : 1B
10B0  A5 00 84 29 72 AE 08 6A  : E4
10B8  1B BF A4 6E FB E0 F0 04  : BB
10C0  95 31 B6 F4 7E E7 C4 D6  : 6F
10C8  90 29 86 9A 61 AE B0 44  : DC
10D0  8E FE 63 6D A1 28 89 1B  : C9
10D8  42 60 88 9C F9 82 BA EB  : E6
10E0  99 D2 8D 7A 56 D1 B9 62  : B4
10E8  5A 33 21 35 23 8B 69 42  : 3C
10F0  58 61 06 CD BC 3E E8 69  : D7
10F8  74 6B 74 C3 F7 3E 6B 34  : EA
-----------------------------------
SUM:  A3 7D 46 1A 57 B1 99 8E  5446

1100  83 F5 36 FE B9 12 FC BA  : 2D
1108  48 3B C6 F2 CA 63 9B 5C  : 5F
1110  8E 51 D5 1A EB 4C 7E 8D  : 10
1118  B1 64 CC 68 DB 60 E8 99  : 05
1120  3A 98 30 35 93 53 78 66  : FB
1128  37 B3 07 13 DA F6 CD C9  : 6A
1130  B3 BE 1F 18 83 C1 C0 35  : E1
1138  62 C6 E8 C2 28 DB 7D D7  : 29
1140  55 A0 2D 75 E0 9F D3 0A  : F3
1148  36 A4 56 2F 02 A0 34 C1  : F6
1150  5C 2A 52 33 7A 28 17 A3  : 67
1158  9C F6 03 E0 EE 76 3E 30  : 47
1160  4A F3 05 CF 8D D4 0C CA  : 48
1168  5D 8E 73 A4 1D 13 53 13  : 98
1170  B5 03 60 00 00 00 00 00  : 18
1178  00 00 00 00 00 00 00 00  : 00
-----------------------------------
SUM:  6F 9C 8B BE 55 CA 3A F2  E762
```

Illustration: T. Takahashi

マシン語カクテル
in Z80's Bar
第35回

今月は，COMETエミュレータならぬCASLのソースプログラムをZ80のソースへ変換するコンバータを制作します。思いつきはよかったけど，仕様の違いから光君はずいぶんと苦労した様子。ダンプリストも掲載しますので皆さんも遊んでみませんか。

お城と流れ星−その2−

Kaneko Shunichi　**金子 俊一**

カランコロ～ン♪
ようこ（以下Yo）：いらっしゃ～い。
源光（以下光）：こんにちは。
Yo：あ～あ，万世（まんせ）だ。
光：何が万世なんですか。
長老（以下老）：肉の万世じゃよ。知らんのか，おぬし。
光：知ってますよ。秋葉原に通ってたころは毎日のように見てましたから。
Yo：それじゃ，そういうことで。
光：なんだかわからないじゃないですか。
Yo：光君さ，先月原稿落としたでしょ。
光：ギクッ！
Yo：数ある連載でも，シリーズもので原稿落とすなんて聞いたことないわよ。
光：申し訳ないですだ。
Yo：ってことで，万世でおごりね。
光：そんな殺生な。
老：まあ身から出たサビというやつじゃろうて。
光：長老もそんなこといわないでくださいよ。せっかくプログラムを作ろうと思ってきたのに。
マスター（以下M）：それじゃ，さっそく作ってもらいましょうかね。
光：っと，その前に，前回純ちゃんは何をやったのかな。
老：ほれ，7月号。
光：ふむふむ。なかなか面白い解析法をやってるね。
Yo：ねえ光君，万世をチャラにしてあげてもいいからさぁ，
光：僕とデートしてほしい，と。
Yo：だれがそんなことぬかすんじゃ，ボケ！
老：なんだか柴田君が書いてるようこちゃんみたいじゃのう。
Yo：CASLのプログラムを実際に動かしてみたいのよ。
光：だったらS-OS用のやつが1986年の12月号に載ってるよ。
Yo：どこにそんなに古いOh!Xがあるってのよ。
老：そのころはまだOh!MZなのじゃが。
Yo：んなこたぁどうでもいいのよ。要はバックナンバーが手に入らないっていってんのよ。
M：それじゃあ作るしかなさそうですね。

エミュレータ，それとも？

光：それじゃあどうやって作ろうかな。
Yo：どうやって，って？
光：一般的には，COMETエミュレータを作るのが普通でしょうね。
老：ほかにあるんかいの。
光：いや，ないですね。
老：それでは悩む必要もあるまいに。さっさとエミュレータを作らんかい。
M：まさかハード音痴の光君がエミュレータを作れるとも考えられませんしねえ。
光：マスター，いってはならないことをいってしまいましたね。
老：死んでもらいます，ってか。
Yo：とにかく作ってよ。
光：いや，実はCASLで書かれたプログラムをZ80のコードに展開できないかなあって考えていたんですよ。
老：ほうほう。
光：PUSHとかCALLとか比較的Z80に似てるから，簡単にできるんじゃないかと。
M：純ちゃんのやり方を見て思いついたんでしょう。
光：するどいですね。
老：やれるんかいのう。
光：ものは試しですよ。ちょっとリストを作ってみましょう。カチャカチャ。
老：無駄にならんといいが。
Yo：夢のない老人は黙ってなさい。
M：今日のようこちゃんは厳しいですね。
光：これを見てください（リスト1）。
老：どれどれ……ほほう。
M：こうやって展開すると不可能ではなさそうですね。
光：それでは，このセンでやってみましょう。

戦慄のメモリ空間

光：しまったあああぁ！
Yo：店の中で何大声はりあげてるのよ。
光：CASLのメモリ空間って知ってます？
Yo：いいえ。
光：あのね，ちょっと奇妙な構成をしているんですよ。
Yo：未来につながっている。
老：モアイが群生している。
M：ピラミッドみたいになっている。
光：みんなでボケかまして。
老：本当はなんなんじゃ。
光：1番地に16ビットが対応しているんですよ。
Yo：ってことは？
光：Z80なんかだったら1番地は8ビットでしょ。
Yo：何か不都合でもあるの？
光：おおありですよ。STとかの命令がおかしくなっちゃう。

Yo：そうなの？

光：たとえば，8000$_{H}$番地にGR0をストアして，8001$_{H}$番地にGR1をストアするってことが実際なら起こりうるんですよ。

Yo：そっか，GR0とかのレジスタって16ビットだもんね。

光：Z80ではメモリ上に重なっちゃうことになるでしょ。

老：まあ適当につじつまを合わせるしかなさそうじゃのう。

光：おそらく一部のプログラムは，そのままでは動かないことになるでしょうね。

フィルタ

光：今回はでっかくなりそうだな。

Yo：まとめてツケが払えるわね。

老：よかったのう。

光：文字列操作をやっているだけなんで，基本は簡単なんですよ。

Yo：データも多そうね。

光：さあて，ぼちぼち完成かな。

Yo：どれどれ？

光：それではサンプルプログラムを組んでみましょう。

Yo：どんなプログラム？

光：えっと，文字列を入力させて，それに改行コード($0D)を加えて出力するだけなんですが（リスト2）。

Yo：変換後はずいぶんと大きくなるのね。

老：無駄な処理もあるようじゃのう。

光：ええ，これでもプログラムを小さくしようと努力したんですけどね。

老：それがかえってCASLからコンバートされたプログラムを大きくするようじゃな。

光：ええ。まあアセンブルするのはアセンブラだし，CASLのプログラムを作ればあとは入力しなくてすみますからね。

老：しようがないのかのう。

M：このプログラムって，強力なマクロをもったエディタなら，マクロだけでもできるかもしれないですね。

光：ただのフィルタだからね。

光流CASL調理講座

M：それではお待ちかねの解説をしてもらいましょうか。

光：まず，CASLで書かれたソースリストをすべてS-OS上の特殊ワークエリアに待避させます。

Yo：特殊ワークエリアが足りなくなることはないの？

光：死ぬほど大きなCASLのプログラムって見たことないから大丈夫だと思う。

Yo：それって上限はないの？

光：約束では256ワード以上のプログラムはないはずだから。

Yo：ふうん。

老：結構メモリを食うようじゃな。

光：それはソースプログラム（リスト3）だけですよ。オブジェクトは4Kバイトもいかないんですから。

老：CASLのソースからZ80へ変換するときだけでいいのじゃな。

光：いえ，実はランタイムルーチンを含んでいますので。

Yo：ランタイムルーチンって何？

光：コンパイラなんかでは常識なんだけど，実際にプログラムを動かすときに必要なルーチン集と思ってください。文字列変換のみでは，ちょっとサポートしきれない命令

リスト1

```
;           CASL TEST ( LD )

;           LD      GR0,$2163

            LD      HL,($2163)
            LD      (GR0),HL

;           LD      GR1,$8000,GR2

            LD      DE,$8000
            LD      HL,(GR2)
            ADD     HL,DE
            LD      E,(HL)
            INC     HL
            LD      D,(HL)
            EX      DE,HL
            LD      (GR1),HL


;           CASL TEST ( ST )

;           ST      GR3,$9801

            LD      HL,(GR3)
            LD      ($9801),HL

;           LEA     GR2,$0001
;           ST      GR1,$9802,GR2

            LD      HL,$0001
            LD      (GR2),HL
            CALL    FLAG
            ;
            LD      DE,$9802
            LD      HL,(GR2)
            ADD     HL,DE
            LD      DE,(GR1)
            LD      (HL),E
            INC     HL
            LD      (HL),D


;           CASL TEST ( LEA )

;           LEA     GR2,$1324

            LD      HL,$1324
            LD      (GR2),HL
            CALL    FLAG

;           LEA     GR0,1,GR2

            LD      DE,1
            LD      HL,(GR2)
            ADD     HL,DE
            LD      (GR0),HL
            CALL    FLAG

;           LEA     GR3,-3,GR0

            LD      DE,-3
            LD      HL,(GR0)
            ADD     HL,DE
            LD      (GR3),HL
            CALL    FLAG


;           CASL TEST ( ADD )

;           LEA     GR0,$1000
;           ADD     GR0,$8366

            LD      HL,$1000
            LD      (GR0),HL
            CALL    FLAG
            ;
            LD      DE,($8366)
            LD      HL,(GR0)
            ADD     HL,DE
            LD      (GR0),HL
            CALL    FLAG

;           LEA     GR1,$1000
;           LEA     GR2,1
;           ADD     GR1,$8367,GR2

            LD      HL,$1000
            LD      (GR1),HL
            CALL    FLAG
            ;
            LD      HL,1
            LD      (GR2),HL
            CALL    FLAG
            ;
            LD      DE,$8367
            LD      HL,(GR2)
            ADD     HL,DE
            LD      E,(HL)
            INC     HL
            LD      D,(HL)
            LD      HL,(GR1)
            ADD     HL,DE
            LD      (GR1),HL
            CALL    FLAG


;           CASL TEST ( SUB )

;           LEA     GR3,$FFFF
;           SUB     GR3,$8368,GR2

            LD      HL,$FFFF
            LD      (GR3),HL
            CALL    FLAG
            ;
            LD      DE,$8368
            LD      HL,(GR2)
            ADD     HL,DE
            LD      E,(HL)
            INC     HL
            LD      D,(HL)
            LD      HL,(GR3)
            OR      A
            SBC     HL,DE
            LD      (GR3),HL
            CALL    FLAG


;           CASL TEST ( AND )

;           LEA     GR0,$1111
;           AND     GR0,$8000

            LD      HL,$1111
            LD      (GR0),HL
            CALL    FLAG
            ;
            LD      DE,($8000)
            LD      HL,(GR0)
            LD      A,L
            AND     E
            LD      L,A
            LD      A,H
            AND     D
            LD      H,A
            LD      (GR0),HL
            CALL    FLAG


;           CASL TEST ( OR )

;           LEA     GR1,0
;           LEA     GR2,1
```

があったのでサブルーチン化してしまいました。

Yo：ふう〜ん。

老：それはどのくらいじゃ。

光：256バイトもありませんよ。DF00$_H$番地からDFBD$_H$番地までですから。

構文チェックはなしよ

老：ところで構文のチェックはやっておるんかいのう。

光：やってません。

M：何かわけありですか。

光：ええ，実は。

Yo：面倒くさかったんでしょ。

M：読まれてますね，光君。

光：だって，レジスタGR0はPUSHはできないのにPOPはできるとか，意味不明の約束事が多すぎるんですよ。

Yo：ほかには省略したこととかあるの？

光：えっと，レジスタGR0を0，レジスタGR1を1のように略して書ける場合があるんですよ。

Yo：うんうん，教科書にも載ってたわ。

老：ようこちゃんは教科書を持っておるのか。

Yo：一応ね。

光：で，話の続きなんですけど，このプログラムでは略して書いたものは判別しません。素直にレジスタGR?って書いてください。

老：まあ，それはプログラムの見やすさからいっても正解じゃろ。

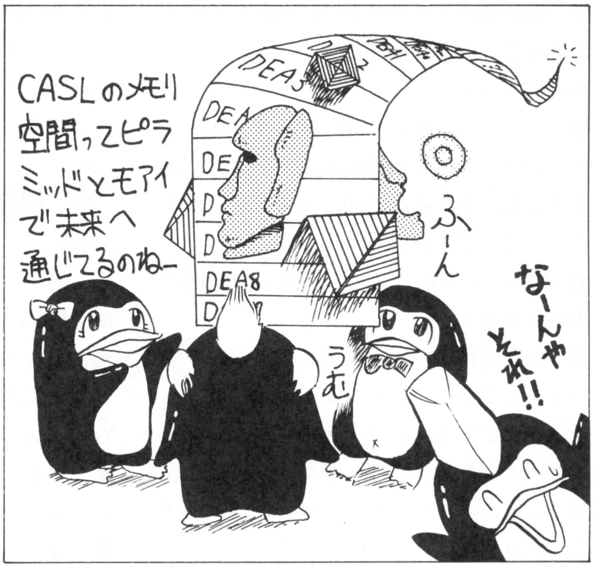
光：「ST 1,$8000,2」よりも「ST GR1,$8000,GR2」のほうがいいですよね。

老：アセンブラのくせに略して書けるなど言語道断じゃわい。

Yo：話は変わるけど，CASLのINとOUTって，ちょっと変わっていたわよね。教科書によっては受け渡すデータをスタックに積むやつもあるし。

光：私は素直に「OUT WORK, LENGTH」「IN WORK, LENGTH」という手法にしました。

老：WORKに文字列の入るアドレスを入れて，その長さをLENGTHに入れるのじゃな。



HOW TO CASL

Yo：最後に動かし方をやってよ。

光：えっとですね。プログラムを正確に打ち込んだら，セーブしておいてください。

Yo：いつものことね。

```
;       OR      GR1,$8000,GR2

        LD      HL,0
        LD      (GR1),HL
        CALL    FLAG
        ;
        LD      HL,1
        LD      (GR2),HL
        CALL    FLAG
        ;
        LD      DE,$8000
        LD      HL,(GR2)
        ADD     HL,DE
        LD      E,(HL)
        INC     HL
        LD      D,(HL)
        LD      HL,(GR1)
        LD      A,L
        OR      E
        LD      L,A
        LD      A,H
        OR      D
        LD      H,A
        LD      (GR1),HL
        CALL    FLAG


;       CASL TEST ( EOR )

;       EOR     GR2,$8002

        LD      DE,($8002)
        LD      HL,(GR2)
        LD      A,L
        XOR     E
        LD      L,A
        LD      A,H
        XOR     D
        LD      H,A
        LD      (GR2),HL
        CALL    FLAG


;       CASL TEST ( JPZ )
        ;
;       JPZ     PLUS

        LD      A,(FR)
        CP      $10
        JP      NZ,PLUS


;       JMI     MINUS

        LD      A,(FR)
        CP      $10
        JP      Z,MINUS


;       JNZ     NZERO

        LD      A,(FR)
        CP      $01
        JP      NZ,NZERO


;       JZE     ZERO

        LD      A,(FR)
        CP      $01
        JP      Z,ZERO


;       JMP     ALL

        JP      ALL

;       JMP     INDEX,GR1

        LD      DE,INDEX
        LD      HL,(GR1)
        SLA     L
        RL      H               ; HL=HL*2
        ADD     HL,DE
        PUSH    HL
        RET                     ; JP HL


;       CASL TEST ( PUSH )

;       PUSH    12345

        LD      HL,12345
        PUSH    HL
        LD      (GR4),SP

;       PUSH    $9000,GR1

        LD      DE,$9000
        LD      HL,(GR1)
        ADD     HL,DE
        PUSH    HL
        LD      (GR4),SP


;       CASL TEST ( POP )

;       POP     GR2

        POP     HL
        LD      (GR2),HL
        LD      (GR4),SP

;       POP     GR3

        POP     HL
        LD      (GR3),HL
        LD      (GR4),SP


;       CASL TEST ( CPL )

;       LEA     GR0,$FFFF
;       CPL     GR0,$9000

        LD      HL,$FFFF
        LD      (GR0),HL
        CALL    FLAG
        ;
        LD      DE,($9000)
        LD      HL,(GR0)
        OR      A
        SBC     HL,DE
        CALL    FLAG_L


;       CASL TEST ( CPA )

;       CPA     GR0,$9000,GR2

        LD      DE,$9000
        LD      HL,(GR2)
        ADD     HL,DE
        LD      E,(HL)
        INC     HL
        LD      D,(HL)
        ;
        LD      HL,(GR0)
        OR      A
        SBC     HL,DE
        CALL    FLAG_A

        RET
```

光：実行には，メモリ上にCASLのソースがあることが前提条件です。

Yo：ふむふむ。

光：そこで，S-OSからJD000とすると，ソースのアドレスと実行開始番地を聞いてきますから，それぞれ5000，7000とかのように答えてください。

老：ほう。CASLのソースリストが画面に出てきたのう。

光：現在変換中の行を表示しているんですけど，そこそこに速いでしょ。

老：そこでアセンブラを起動すれば，アセンブルできるようになっておるのじゃな。

光：ええ，エラーがあった場合にはビープ音が鳴ります。

老：新しいソースはどこに作られるのじゃ。

光：もとのCASLのソースがあった番地に上書きされます。

老：ということは，ここでエディタを起動すれば，すぐエディットできるわけか。

光：ええ。慣れるとなかなか快適ですよ。

最後に

老：何かいい残したことはないかのう。

M：長老，縁起でもないですよ。

光：えっと，リスト2に注釈でZ80アレンジ，って書いてある行がありますよね。

Yo：ええ，あるわ。

光：あそこは，本来CASLだったらいらない行なんですよ。

老：ほほう。

光：1番地が16ビットっていうメモリ構成だから，そのままでは実行できなかったんで，アドレスの計算が正しくなるようにしたんです。

老：つまり，そのままでは動かないプログラムもあるってことじゃな。

光：ええ，今回のように簡単な追加で動くようになると思いますけどね。

Yo：完全に，動くプログラムもあるわよね。

光：ええ，算術のルーチンとか，メモリに待避するデータがない場合は完璧に動くはずですよ。具体的に動くもの，動かないものは来月号でお話しします。

老：ほかにはないんか。

光：えっとタブコードに対応していません。

老：というと？

光：僕が使っているシステムではタブコード対応のソフトがないんですよ。それで試せなかったもんだから。

リスト2

```
;       TEST of IN & OUT  (ADD LTNL)
;
;       Hikaru Minamoto
        START
        IN      WORD,LENG
        ;
        LEA     GR0,WORD
        LD      GR1,LENG
        ADD     GR0,GR1
        ADD     GR0,GR1    ; Z80 ARRANGE
        ;
        LEA     GR2,$0D
        ST      GR2,0,GR0
        ;
        LEA     GR1,1,GR1
        ST      GR1,LENG
        ;
        OUT     WORD,LENG
        END
LENG
        DS      1
WORD
        DS      80
```

リスト3

```
7000           1          ORG     $7000
7000           2
7000           3 REG      EQU     $DF00
7000           4 GR0      EQU     REG
7000           5 GR1      EQU     REG+2
7000           6 GR2      EQU     REG+4
7000           7 GR3      EQU     REG+6
7000           8 GR4      EQU     REG+8
7000           9 FR       EQU     REG+10
7000          10 OUT      EQU     REG+11
7000          11 IN       EQU     REG+13
7000          12 SLL      EQU     REG+15
7000          13 SRL      EQU     REG+17
7000          14 SLA      EQU     REG+19
7000          15 SRA      EQU     REG+21
7000          16 SHIFT    EQU     REG+23
7000          17 FLAG_A   EQU     REG+25
7000          18 FLAG_L   EQU     REG+28
7000          19 FLAG     EQU     REG+30
7000          20
7000          21 ;;          TEST of IN & OUT  (ADD LTNL)
7000          22 ;;
7000          23 ;;          Hikaru Minamoto
7000          24 ;
7000          25 ;           START
7000          26 ;           IN       WORD,LENG
7000 21 6A 70 27               LD       HL,WORD
7003 01 68 70 28               LD       BC,LENG
7006 CD 0D DF 29               CALL     IN
7009          30 ;           ;
7009          31 ;           LEA      GR0,WORD
7009 21 6A 70 32               LD       HL,WORD
700C 22 00 DF 33               LD       (GR0),HL
700F CD 1E DF 34               CALL     FLAG
7012          35 ;           LD       GR1,LENG
7012 2A 68 70 36               LD       HL,(LENG)
7015 22 02 DF 37               LD       (GR1),HL
7018          38 ;           ADD      GR0,GR1
7018 2A 02 DF 39               LD       HL,(GR1)
701B ED 5B 00 40               LD       DE,(GR0)
701E DF
701F 19       41               ADD      HL,DE
7020 22 00 DF 42               LD       (GR0),HL
7023 CD 1E DF 43               CALL     FLAG
7026          44 ;           ADD      GR0,GR1    ; Z80 ARRANGE
7026 2A 02 DF 45               LD       HL,(GR1)
7029 ED 5B 00 46               LD       DE,(GR0)
702C DF
702D 19       47               ADD      HL,DE
702E 22 00 DF 48               LD       (GR0),HL
7031 CD 1E DF 49               CALL     FLAG
7034          50 ;           ;
7034          51 ;           LEA      GR2,$0D
7034 21 0D 00 52               LD       HL,$0D
7037 22 04 DF 53               LD       (GR2),HL
703A CD 1E DF 54               CALL     FLAG
703D          55 ;           ST       GR2,0,GR0
703D 11 00 00 56               LD       DE,0
7040 2A 00 DF 57               LD       HL,(GR0)
7043 19       58               ADD      HL,DE
7044 ED 5B 04 59               LD       DE,(GR2)
7047 DF
7048 73       60               LD       (HL),E
7049 23       61               INC      HL
704A 72       62               LD       (HL),D
704B          63 ;           ;
704B          64 ;           LEA      GR1,1,GR1
704B 11 01 00 65               LD       DE,1
704E 2A 02 DF 66               LD       HL,(GR1)
7051 19       67               ADD      HL,DE
7052 22 02 DF 68               LD       (GR1),HL
7055 CD 1E DF 69               CALL     FLAG
7058          70 ;           ST       GR1,LENG
7058 2A 02 DF 71               LD       HL,(GR1)
705B 22 68 70 72               LD       (LENG),HL
705E          73 ;           ;
705E          74 ;           OUT      WORD,LENG
705E 21 6A 70 75               LD       HL,WORD
7061 01 68 70 76               LD       BC,LENC
7064 CD 0B DF 77               CALL     OUT
7067          78 ;           END
7067 C9       79               RET
7068          80 ;LENG
7068          81 LENG
7068          82 ;           DS       1
7068 00 00    83               DS       1*2
706A          84 ;WORD
706A          85 WORD
706A          86 ;           DS       80
706A 00 00 00 87               DS       80*2
```

閉店のお時間

M：今回はリストが長かったですね。

光：そうでしょ，だてに1カ月も待たせたわけじゃないでしょ。

Yo：そうやって逃げる。

M：まあよしとしましょう。

Yo：ところで光君って，情報処理の資格もってるの？

光：はっはっは。なぜか1種を。

一同：げげっ！

光：勉強してなかったんですけどね。前の日も飲み会でさっさと寝ちゃったし。

M：CASLのほかにはどの言語で受けました？

光：FORTRAN。

老：おぬしFORTRANも知っておるのか。

光：試験当日に電車の中で勉強しただけです。ちなみにCASLを勉強したのは2年くらい前だったかな。

M：そんなんで受かっちゃうんですか。

光：奇跡はあるのです。

老：世の中は不公平にできておるのう。

Yo：人のノートでAを取る男はここでも健在だったのね。

―つづく―

リスト4

```
0000               1 ;       CASL Transmitter
0000               2 ;
0000               3 ;          by Hikaru Minamoto
0000               4
D000               5           ORG     $D000
D000               6
D000               7 ;Label             Address  Break
D000               8
D000               9 [HL]      EQU     $1F81   ; nothing
D000              10 #PEEK     EQU     $1F94   ; AF
D000              11 #POKE     EQU     $1F9A   ; nothing
D000              12 #HLHEX    EQU     $1FB2   ; AF,DE+4,HL
D000              13 #ASC      EQU     $1FBB   ; AF
D000              14 #BELL     EQU     $1FC4   ; AF
D000              15 #PAUSE    EQU     $1FC7   ; AF
D000              16 #GETL     EQU     $1FD3   ; AF
D000              17 #MPRINT   EQU     $1FE2   ; AF,DE
D000              18 #MSG      EQU     $1FE8   ; F
D000              19 #LTNL     EQU     $1FEE   ; nothing
D000              20 #PRINT    EQU     $1FF4   ; F
D000              21
D000              22 PRE
D000 CD E2 1F     23           CALL    #MPRINT
D003 54 45 58     24           DM      "TEXT Address = $"
D006 54 20 41
D009 64 64 72
D00C 65 73 73
D00F 20 3D 20
D012 24
D013 00           25           DB      0
D014 CD 37 D0     26           CALL    GETADR
D017 22 42 D9     27           LD      (TXTADR),HL
D01A              28           ;
D01A CD E2 1F     29           CALL    #MPRINT
D01D 45 58 45     30           DM      "EXEC Address = $"
D020 43 20 41
D023 64 64 72
D026 65 73 73
D029 20 3D 20
D02C 24
D02D 00           31           DB      0
D02E CD 37 D0     32           CALL    GETADR
D031 22 40 D9     33           LD      (EXEADR),HL
D034 C3 6F D0     34           JP      WORKSET
D037              35 GETADR
D037 11 DB DA     36           LD      DE,KEYBUF
D03A CD D3 1F     37           CALL    #GETL
D03D              38           ;
D03D 1A           39           LD      A,(DE)
D03E FE 1B        40           CP      $1B
D040 CA 5E D0     41           JP      Z,END2
D043              42           ;
D043 11 EB DA     43           LD      DE,KEYBUF+$10
D046 CD B2 1F     44           CALL    #HLHEX
D049 D0           45           RET     NC
D04A CD E2 1F     46           CALL    #MPRINT
D04D 53 79 6E     47           DM      "Syntax Error."
D050 74 61 78
D053 20 45 72
D056 72 6F 72
D059 2E
D05A 00           48           DB      0
D05B CD EE 1F     49           CALL    #LTNL
D05E              50 END2
D05E 2A 42 D9     51           LD      HL,(TXTADR)
D061 AF           52           XOR     A
D062 77           53           LD      (HL),A
D063              54           ;
D063 CD E2 1F     55           CALL    #MPRINT
D066 45 4E 44     56           DM      "END."
D069 2E
D06A 00           57           DB      0
D06B CD EE 1F     58           CALL    #LTNL
D06E C9           59           RET
D06F              60           ;
D06F              61 WORKSET
D06F 2A 42 D9     62           LD      HL,(TXTADR)
D072 11 00 00     63           LD      DE,$0000
D075 ED 53 44     64           LD      (WRKADR),DE
D078 D9
D079              65 WRKSET2
D079 7E           66           LD      A,(HL)
D07A EB           67           EX      DE,HL
D07B CD 9A 1F     68           CALL    #POKE
D07E EB           69           EX      DE,HL
D07F 13           70           INC     DE
D080 23           71           INC     HL
D081 B7           72           OR      A
D082 20 F5        73           JR      NZ,WRKSET2
D084              74           ;
D084 ED 5B 42     75           LD      DE,(TXTADR)
D087 D9
D088 21 48 D9     76           LD      HL,ORG$
D08B 01 11 00     77           LD      BC,17
D08E ED B0        78           LDIR
D090              79           ;
D090 EB           80           EX      DE,HL
D091 3A 41 D9     81           LD      A,(EXEADR+1)
D094 CD AC D0     82           CALL    HLASC
D097 3A 40 D9     83           LD      A,(EXEADR)
D09A CD AC D0     84           CALL    HLASC
D09D              85           ;
D09D EB           86           EX      DE,HL
D09E 21 59 D9     87           LD      HL,OYAKUSOKU
D0A1 01 82 01     88           LD      BC,OYA2-OYAKUSOKU
D0A4 ED B0        89           LDIR
D0A6 ED 53 42     90           LD      (TXTADR),DE
D0A9 D9
D0AA 18 11        91           JR      GETL
D0AC              92 HLASC
D0AC F5           93           PUSH    AF
D0AD 0F           94           RRCA
D0AE 0F           95           RRCA
D0AF 0F           96           RRCA
D0B0 0F           97           RRCA
D0B1 CD BB 1F     98           CALL    #ASC
D0B4 77           99           LD      (HL),A
D0B5 23          100           INC     HL
D0B6 F1          101           POP     AF
D0B7 CD BB 1F    102           CALL    #ASC
D0BA 77          103           LD      (HL),A
D0BB 23          104           INC     HL
D0BC C9          105           RET
D0BD             106 GETL
D0BD 11 DB DA    107           LD      DE,KEYBUF
D0C0 2A 44 D9    108           LD      HL,(WRKADR)
D0C3             109 GETL2
D0C3 CD 94 1F    110           CALL    #PEEK
D0C6 B7          111           OR      A
D0C7 28 95       112           JR      Z,END2
D0C9             113           ;
D0C9 12          114           LD      (DE),A
D0CA 13          115           INC     DE
D0CB 23          116           INC     HL
D0CC FE 0D       117           CP      $0D
D0CE 28 02       118           JR      Z,XASS
D0D0 18 F1       119           JR      GETL2
D0D2             120 XASS
D0D2 22 44 D9    121           LD      (WRKADR),HL
D0D5 11 DB DA    122           LD      DE,KEYBUF
D0D8 CD E8 1F    123           CALL    #MSG
D0DB CD EE 1F    124           CALL    #LTNL
D0DE CD E8 D0    125           CALL    RECOPY
D0E1 CD C7 1F    126           CALL    #PAUSE
D0E4 5E D0       127           DW      END2
D0E6 18 D5       128           JR      GETL
D0E8             129 RECOPY
D0E8 2A 42 D9    130           LD      HL,(TXTADR)
D0EB 3E 3B       131           LD      A,";"
D0ED 77          132           LD      (HL),A
D0EE 23          133           INC     HL
D0EF             134 RECOPY2
D0EF 1A          135           LD      A,(DE)
D0F0 77          136           LD      (HL),A
D0F1 13          137           INC     DE
D0F2 23          138           INC     HL
D0F3 FE 0D       139           CP      $0D
D0F5 20 F8       140           JR      NZ,RECOPY2
D0F7 22 42 D9    141           LD      (TXTADR),HL
D0FA             142           ;
D0FA             143 PASER
D0FA 11 DB DA    144           LD      DE,KEYBUF
D0FD 1A          145           LD      A,(DE)
D0FE FE 20       146           CP      " "
D100 28 18       147           JR      Z,NOLABEL
D102 FE 0D       148           CP      $0D
```

```
D104 28 14        149         JR      Z,NOLABEL
D106 FE 3B        150         CP      ";"
D108 28 7D        151         JR      Z,RETXASS
D10A              152 LABEL
D10A 1A           153         LD      A,(DE)
D10B 77           154         LD      (HL),A  ; LABEL COPY
D10C 23           155         INC     HL
D10D 13           156         INC     DE
D10E FE 0D        157         CP      $0D
D110 28 75        158         JR      Z,RETXASS
D112 FE 20        159         CP      " "
D114 20 F4        160         JR      NZ,LABEL
D116 CD 10 D6     161         CALL    LTNL$
D119 AF           162         XOR     A       ; DUMMY
D11A              163 NOLABEL
D11A FE 0D        164         CP      $0D     ; SPACE CUT
D11C 28 69        165         JR      Z,RETXASS
D11E 1A           166         LD      A,(DE)
D11F FE 20        167         CP      " "
D121 13           168         INC     DE
D122 28 F6        169         JR      Z,NOLABEL
D124              170         ;
D124 FE 3B 20     171         IF A=";" THEN JP RETXASS
D127 03 C3 87
D12A D1
D12B F5           172         PUSH    AF
D12C CD 37 D6     173         CALL    TAB11
D12F F1           174         POP     AF
D130              175         ;
D130 FE 41 20     176         IF A="A" THEN JP A_PASER
D133 03 C3 8B
D136 D1
D137 FE 43 20     177         IF A="C" THEN JP C_PASER
D13A 03 C3 F5
D13D D1
D13E FE 44 20     178         IF A="D" THEN JP D_PASER
D141 03 C3 29
D144 D2
D145 FE 45 20     179         IF A="E" THEN JP E_PASER
D148 03 C3 AA
D14B D2
D14C FE 49 20     180         IF A="I" THEN JP I_PASER
D14F 03 C3 D5
D152 D2
D153 FE 4A 20     181         IF A="J" THEN JP J_PASER
D156 03 C3 DB
D159 D2
D15A FE 4C 20     182         IF A="L" THEN JP L_PASER
D15D 03 C3 AE
D160 D3
D161 FE 4F 20     183         IF A="O" THEN JP O_PASER
D164 03 C3 20
D167 D4
D168 FE 50 20     184         IF A="P" THEN JP P_PASER
D16B 03 C3 40
D16E D4
D16F FE 52 20     185         IF A="R" THEN JP R_PASER
D172 03 C3 87
D175 D4
D176 FE 53 20     186         IF A="S" THEN JP S_PASER
D179 03 C3 90
D17C D4
D17D              187         ;
D17D              188 ERR
D17D CD C4 1F     189         CALL    #BELL
D180 3E 3F        190         LD      A,"?"   ; ERROR !!
D182 77           191         LD      (HL),A
D183 23           192         INC     HL
D184 CD 10 D6     193         CALL    LTNL$
D187              194 RETXASS
D187 22 42 D9     195         LD      (TXTADR),HL
D18A C9           196         RET
D18B              197 A_PASER
D18B 1A           198         LD      A,(DE)
D18C FE 44 20     199         IF A="D" THEN JR ADD_SET
D18F 02 18 13
D192 FE 4E 20     200         IF A="N" THEN JR AND_SET
D195 02 18 03
D198 C3 7D D1     201         JP      ERR
D19B              202 AND_SET
D19B 01 32 D9     203         LD      BC,AND$
D19E ED 43 46     204         LD      (LOG_WRK),BC
D1A1 D9
D1A2 C3 9B D6     205         JP      LOG_SET
D1A5              206 ADD_SET
D1A5 13           207         INC     DE      ;D
D1A6              208         ;
D1A6 CD A2 D5     209         CALL    SPCCUT
D1A9 D5           210         PUSH    DE      ;G
D1AA 13           211         INC     DE      ;R
D1AB 13           212         INC     DE      ;?
D1AC 13           213         INC     DE      ;,
D1AD 13           214         INC     DE      ;?
D1AE CD A9 D5     215         CALL    CCHECK
D1B1 01 C2 D1     216         LD      BC,ADD_ST2
D1B4 C5           217         PUSH    BC
D1B5 D2 D6 D5     218         JP      NC,LDHL2$
D1B8 C1           219         POP     BC
D1B9 CD EA D5     220         CALL    INDEX$
D1BC 11 5F D8     221         LD      DE,LD$3
D1BF CD 1A D6     222         CALL    SET$
D1C2              223 ADD_ST2
D1C2 CD 37 D6     224         CALL    TAB11
D1C5 11 85 D7     225         LD      DE,LDE2s
D1C8 CD 1A D6     226         CALL    SET$
D1CB D1           227         POP     DE
D1CC D5           228         PUSH    DE
D1CD 01 03 00     229         LD      BC,3
D1D0 CD 15 D6     230         CALL    TRNS$
D1D3 11 EC D7     231         LD      DE,IND$3
D1D6 CD 1A D6     232         CALL    SET$
D1D9 11 47 D8     233         LD      DE,LD$
D1DC CD 1A D6     234         CALL    SET$
D1DF D1           235         POP     DE
D1E0 01 03 00     236         LD      BC,3
D1E3 CD 15 D6     237         CALL    TRNS$
D1E6 11 4D D8     238         LD      DE,LD$2
D1E9 CD 1A D6     239         CALL    SET$
D1EC 11 53 D8     240         LD      DE,CALL_FLAG
D1EF CD 1A D6     241         CALL    SET$
D1F2 C3 87 D1     242         JP      RETXASS
D1F5              243 C_PASER
D1F5 1A           244         LD      A,(DE)
D1F6 FE 41 20     245         IF A="A" THEN JR CALL_SET
D1F9 02 18 09
D1FC FE 50 20     246         IF A="P" THEN JR CP_SET
D1FF 02 18 0A
D202 C3 7D D1     247         JP      ERR
D205              248 CALL_SET
D205 1B           249         DEC     DE      ;DE=C
D206 CD BF D5     250         CALL    COPY$
D209 C3 87 D1     251         JP      RETXASS
D20C              252 CP_SET
D20C 13           253         INC     DE
D20D 1A           254         LD      A,(DE)
D20E FE 4C 20     255         IF A="L" THEN JR CPL_SET
D211 02 18 09
D214 FE 41 20     256         IF A="A" THEN JR CPA_SET
D217 02 18 09
D21A C3 7D D1     257         JP      ERR
D21D              258 CPL_SET
D21D 01 60 D7     259         LD      BC,CPL$
D220 C3 3B D6     260         JP      CPAL_SET
D223              261 CPA_SET
D223 01 52 D7     262         LD      BC,CPA$
D226 C3 3B D6     263         JP      CPAL_SET
D229              264 D_PASER
D229 1A           265         LD      A,(DE)
D22A FE 43 20     266         IF A="C" THEN JR DC_SET
D22D 02 18 09
D230 FE 53 20     267         IF A="S" THEN JR DS_SET
D233 02 18 5C
D236 C3 7D D1     268         JP      ERR
D239              269 DC_SET
D239 CD A2 D5     270         CALL    SPCCUT
D23C D5           271         PUSH    DE
D23D 1A           272         LD      A,(DE)
D23E FE 23 20     273         IF A="#" THEN JR DC_ST3
D241 02 18 13
D244 FE 22 20     274         IF A=""" THEN JR DC_ST4
D247 02 18 1F
D24A              275 DC_ST2
D24A 11 72 D7     276         LD      DE,DW$
D24D CD 1A D6     277         CALL    SET$
D250 D1           278         POP     DE
D251 CD BF D5     279         CALL    COPY$
D254 C3 87 D1     280         JP      RETXASS
D257              281 DC_ST3
D257 11 72 D7     282         LD      DE,DW$
D25A CD 1A D6     283         CALL    SET$
D25D D1           284         POP     DE
D25E 3E 24        285         LD      A,"$"
D260 77           286         LD      (HL),A
D261 13           287         INC     DE
D262 23           288         INC     HL
D263 CD BF D5     289         CALL    COPY$
D266 C3 87 D1     290         JP      RETXASS
D269              291 DC_ST4
D269 11 0B 00     292         LD      DE,11
D26C B7           293         OR      A
D26D ED 52        294         SBC     HL,DE   ;DEL_TAB
D26F              295 DC_ST5
D26F D1           296         POP     DE
D270 13           297         INC     DE
D271 1A           298         LD      A,(DE)
D272 FE 22        299         CP      """
D274 CA 87 D1     300         JP      Z,RETXASS
D277 D5           301         PUSH    DE
D278 F5           302         PUSH    AF
D279 CD 37 D6     303         CALL    TAB11
D27C 11 72 D7     304         LD      DE,DW$
D27F CD 1A D6     305         CALL    SET$
D282 3E 22        306         LD      A,"""
D284 77           307         LD      (HL),A
D285 23           308         INC     HL
D286 F1           309         POP     AF
D287 77           310         LD      (HL),A
D288 23           311         INC     HL
D289 3E 22        312         LD      A,"""
D28B 77           313         LD      (HL),A
D28C 23           314         INC     HL
D28D CD 10 D6     315         CALL    LTNL$
D290 18 DD        316         JR      DC_ST5
D292              317 DS_SET
D292 CD A2 D5     318         CALL    SPCCUT
D295 D5           319         PUSH    DE
D296 11 6E D7     320         LD      DE,DS$
D299 CD 1A D6     321         CALL    SET$
D29C D1           322         POP     DE
D29D CD BF D5     323         CALL    COPY$
D2A0 2B           324         DEC     HL      ;$0D
D2A1 11 2E D9     325         LD      DE,TIMES2$
```


▶８月号の特集でフローチャートを取り上げていましたが，フローチャートを書く人ってどのくらいいるのでしょうか。僕は，PADのほうが好きです。といっても，いつも何も書きませんけど。

黒武者　健一(22)神奈川県

```
D2A4 CD 1A D6            CALL    SET$
D2A7 C3 87 D1            JP      RETXASS
D2AA           E_PASER
D2AA 1A                  LD      A,(DE)
D2AB FE 4E 20            IF A="N" THEN JP END_SET
D2AE 03 C3 C2
D2B1 D2
D2B2 FE 4F 20            IF A="O" THEN JR EOR_SET
D2B5 02 18 13
D2B8 FE 58 20            IF A="X" THEN JP END_SET
D2BB 03 C3 C2
D2BE D2
D2BF C3 7D D1            JP      ERR
D2C2           END_SET
D2C2 11 BF D8            LD      DE,RET$
D2C5 CD 1A D6            CALL    SET$
D2C8 C3 87 D1            JP      RETXASS
D2CB           EOR_SET
D2CB 01 3B D9            LD      BC,XOR$
D2CE ED 43 46            LD      (LOG_WRK),BC
D2D1 D9
D2D2 C3 9B D6            JP      LOG_SET
D2D5           I_PASER
D2D5 01 C6 D7            LD      BC,IN$
D2D8 C3 78 D6            JP      IO_SET
D2DB           J_PASER
D2DB 1A                  LD      A,(DE)
D2DC FE 5A 20            IF A="Z" THEN JR JZE_SET
D2DF 02 18 39
D2E2 FE 4E 20            IF A="N" THEN JR JNZ_SET
D2E5 02 18 4E
D2E8 FE 50 20            IF A="P" THEN JR JPZ_SET
D2EB 02 18 63
D2EE FE 4D 28            IF A<>"M" THEN JP ERR
D2F1 03 C3 7D
D2F4 D1
D2F5 13                  INC     DE
D2F6 1A                  LD      A,(DE)
D2F7 FE 50 20            IF A="P" THEN JR JMP_SET
D2FA 02 18 73
D2FD           JMI_SET
D2FD CD A2 D5            CALL    SPCCUT
D300 D5                  PUSH    DE
D301 11 FA D7            LD      DE,JP$
D304 CD 1A D6            CALL    SET$
D307 3E 30               LD      A,"0"
D309 77                  LD      (HL),A
D30A 23                  INC     HL
D30B CD 10 D6            CALL    LTNL$
D30E 11 38 D8            LD      DE,JPZ$
D311 CD 1A D6            CALL    SET$
D314 D1                  POP     DE
D315 CD BF D5            CALL    COPY$
D318 C3 87 D1            JP      RETXASS
D31B           JZE_SET
D31B 13                  INC     DE
D31C CD A2 D5            CALL    SPCCUT
D31F D5                  PUSH    DE
D320 11 FA D7            LD      DE,JP$
D323 CD 1A D6            CALL    SET$
D326 CD 10 D6            CALL    LTNL$
D329 11 38 D8            LD      DE,JPZ$
D32C CD 1A D6            CALL    SET$
D32F D1                  POP     DE
D330 CD BF D5            CALL    COPY$
D333 C3 87 D1            JP      RETXASS
D336           JNZ_SET
D336 13                  INC     DE
D337 CD A2 D5            CALL    SPCCUT
D33A D5                  PUSH    DE
D33B 11 FA D7            LD      DE,JP$
D33E CD 1A D6            CALL    SET$
D341 CD 10 D6            CALL    LTNL$
D344 11 3F D8            LD      DE,JPNZ$
D347 CD 1A D6            CALL    SET$
D34A D1                  POP     DE
D34B CD BF D5            CALL    COPY$
D34E C3 87 D1            JP      RETXASS
D351           JPZ_SET
D351 13                  INC     DE
D352 CD A2 D5            CALL    SPCCUT
D355 D5                  PUSH    DE
D356 11 FA D7            LD      DE,JP$
D359 CD 1A D6            CALL    SET$
D35C 3E 30               LD      A,"0"
D35E 77                  LD      (HL),A
D35F 23                  INC     HL
D360 CD 10 D6            CALL    LTNL$
D363 11 3F D8            LD      DE,JPNZ$
D366 CD 1A D6            CALL    SET$
D369 D1                  POP     DE
D36A CD BF D5            CALL    COPY$
D36D C3 87 D1            JP      RETXASS
D370           JMP_SET
D370 CD A2 D5            CALL    SPCCUT
D373 CD A9 D5            CALL    CCHECK
D376 38 0E               JR      C,JP_INDEX
D378 D5                  PUSH    DE
D379 11 34 D8            LD      DE,JMP$
D37C CD 1A D6            CALL    SET$
D37F D1                  POP     DE
D380 CD BF D5            CALL    COPY$
D383 C3 87 D1            JP      RETXASS
D386           JP_INDEX
D386 D5                  PUSH    DE
D387 11 DB D7            LD      DE,IND$
D38A CD 1A D6            CALL    SET$
D38D D1                  POP     DE
D38E           JP_IND2
D38E 1A                  LD      A,(DE)
D38F 77                  LD      (HL),A
D390 13                  INC     DE
D391 23                  INC     HL
D392 FE 2C               CP      ","
D394 20 F8               JR      NZ,JP_IND2
D396 2B                  DEC     HL
D397 D5                  PUSH    DE
D398 11 E2 D7            LD      DE,IND$2
D39B CD 1A D6            CALL    SET$
D39E D1                  POP     DE
D39F 01 03 00            LD      BC,3
D3A2 CD 15 D6            CALL    TRNS$
D3A5 11 0B D8            LD      DE,JP_IND$
D3A8 CD 1A D6            CALL    SET$
D3AB C3 87 D1            JP      RETXASS
D3AE           L_PASER
D3AE 1A                  LD      A,(DE)
D3AF FE 44 20            IF A="D" THEN JR LD_SET
D3B2 02 18 09
D3B5 FE 45 20            IF A="E" THEN JR LEA_SET
D3B8 02 18 35
D3BB C3 7D D1            JP      ERR
D3BE           LD_SET
D3BE CD A2 D5            CALL    SPCCUT
D3C1 D5                  PUSH    DE      ;G
D3C2 13                  INC     DE      ;R
D3C3 13                  INC     DE      ;?
D3C4 13                  INC     DE      ;,
D3C5 13                  INC     DE      ;?
D3C6 CD A9 D5            CALL    CCHECK
D3C9 01 DA D3            LD      BC,LD_ST2
D3CC C5                  PUSH    BC
D3CD D2 D6 D5            JP      NC,LDHL2$
D3D0 C1                  POP     BC
D3D1 CD EA D5            CALL    INDEX$
D3D4 11 5F D8            LD      DE,LD$3
D3D7 CD 1A D6            CALL    SET$
D3DA           LD_ST2
D3DA 11 47 D8            LD      DE,LD$
D3DD CD 1A D6            CALL    SET$
D3E0 D1                  POP     DE
D3E1 01 03 00            LD      BC,3
D3E4 CD 15 D6            CALL    TRNS$
D3E7 11 4D D8            LD      DE,LD$2
D3EA CD 1A D6            CALL    SET$
D3ED C3 87 D1            JP      RETXASS
D3F0           LEA_SET
D3F0 13                  INC     DE      ;A
D3F1                     ;
D3F1 CD A2 D5            CALL    SPCCUT
D3F4 D5                  PUSH    DE      ;G
D3F5 13                  INC     DE      ;R
D3F6 13                  INC     DE      ;?
D3F7 13                  INC     DE      ;,
D3F8 13                  INC     DE      ;?
D3F9 CD A9 D5            CALL    CCHECK
D3FC 01 07 D4            LD      BC,LA_ST2
D3FF C5                  PUSH    BC
D400 D2 B7 D5            JP      NC,LDHL$
D403 C1                  POP     BC
D404 CD EA D5            CALL    INDEX$
D407           LA_ST2
D407 11 47 D8            LD      DE,LD$
D40A CD 1A D6            CALL    SET$
D40D D1                  POP     DE
D40E 01 03 00            LD      BC,3
D411 CD 15 D6            CALL    TRNS$
D414 11 4D D8            LD      DE,LD$2
D417 CD 1A D6            CALL    SET$
D41A CD 1A D6            CALL    SET$    ;CALL_FLAG
D41D C3 87 D1            JP      RETXASS
D420           O_PASER
D420 1A                  LD      A,(DE)
D421 FE 52 20            IF A="R" THEN JR OR_SET
D424 02 18 09
D427 FE 55 20            IF A="U" THEN JR OUT_SET
D42A 02 18 0D
D42D C3 7D D1            JP      ERR
D430           OR_SET
D430 01 37 D9            LD      BC,OR$
D433 ED 43 46            LD      (LOG_WRK),BC
D436 D9
D437 C3 9B D6            JP      LOG_SET
D43A           OUT_SET
D43A 01 D0 D7            LD      BC,OUT$
D43D C3 78 D6            JP      IO_SET
D440           P_PASER
D440 1A                  LD      A,(DE)
D441 FE 55 20            IF A="U" THEN JR PUSH_SET
D444 02 18 09
D447 FE 4F 20            IF A="O" THEN JR POP_SET
D44A 02 18 1F
D44D C3 7D D1            JP      ERR
D450           PUSH_SET
D450 13                  INC     DE      ;S
D451 13                  INC     DE      ;H
D452                     ;
D452 CD A2 D5            CALL    SPCCUT
D455 CD A9 D5            CALL    CCHECK
D458 01 63 D4            LD      BC,PH_ST2
D45B C5                  PUSH    BC
D45C D2 B7 D5            JP      NC,LDHL$
```

▶ microOdysseyにあったUさんの文章が格好ええなあ。全然面識のない人の文章を読むと，文章を書いている人の姿が頭の中に浮かんできませんか。Uさんなんかは“メガネがきらりと光るニヒルなお父さん”という感じだなあ。やっぱり頭が薄い，というのはないだろうな（薄かったらごめんなさい）。　長谷川　聖(28)香川県

```
D45F C1                  POP     BC
D460 CD EA D5            CALL    INDEX$
D463            PH_ST2
D463 11 88 D8            LD      DE,PUSH$
D466 CD 1A D6            CALL    SET$
D469 C3 87 D1            JP      RETXASS
D46C            POP_SET
D46C D5                  PUSH    DE      ;O
D46D 11 9F D8            LD      DE,POP$
D470 CD 1A D6            CALL    SET$
D473 D1                  POP     DE
D474 13                  INC     DE      ;P
D475                     ;
D475 CD A2 D5            CALL    SPCCUT
D478                     ;
D478 01 03 00            LD      BC,3
D47B CD 15 D6            CALL    TRNS$
D47E 11 AC D8            LD      DE,POP$2
D481 CD 1A D6            CALL    SET$
D484 C3 87 D1            JP      RETXASS
D487            R_PASER
D487 11 BF D8            LD      DE,RET$
D48A CD 1A D6            CALL    SET$
D48D C3 87 D1            JP      RETXASS
D490            S_PASER
D490 1A                  LD      A,(DE)
D491 FE 55 20            IF A="U" THEN JR SUB_SET
D494 02 18 17
D497 FE 54 20            IF A="T" THEN JR ST_SET
D49A 02 18 61
D49D FE 52 20            IF A="R" THEN JP SR_SET
D4A0 03 C3 68
D4A3 D5
D4A4 FE 4C 20            IF A="L" THEN JP SL_SET
D4A7 03 C3 85
D4AA D5
D4AB C3 7D D1            JP      ERR
D4AE            SUB_SET
D4AE 13                  INC     DE      ;B
D4AF                     ;
D4AF CD A2 D5            CALL    SPCCUT
D4B2 D5                  PUSH    DE      ;G
D4B3 13                  INC     DE      ;R
D4B4 13                  INC     DE      ;?
D4B5 13                  INC     DE      ;,
D4B6 13                  INC     DE      ;?
D4B7 CD A9 D5            CALL    CCHECK
D4BA 01 CB D4            LD      BC,SUB_ST2
D4BD C5                  PUSH    BC
D4BE D2 D6 D5            JP      NC,LDHL2$
D4C1 C1                  POP     BC
D4C2 CD EA D5            CALL    INDEX$
D4C5 11 5F D8            LD      DE,LD$3
D4C8 CD 1A D6            CALL    SET$
D4CB            SUB_ST2
D4CB CD 37 D6            CALL    TAB11
D4CE 11 85 D7            LD      DE,LDE2$
D4D1 CD 1A D6            CALL    SET$
D4D4 D1                  POP     DE
D4D5 D5                  PUSH    DE
D4D6 01 03 00            LD      BC,3
D4D9 CD 15 D6            CALL    TRNS$
D4DC 11 FC D8            LD      DE,SUB$
D4DF CD 1A D6            CALL    SET$
D4E2 11 47 D8            LD      DE,LD$
D4E5 CD 1A D6            CALL    SET$
D4E8 D1                  POP     DE
D4E9 01 03 00            LD      BC,3
D4EC CD 15 D6            CALL    TRNS$
D4EF 11 4D D8            LD      DE,LD$2
D4F2 CD 1A D6            CALL    SET$
D4F5 11 53 D8            LD      DE,CALL_FLAG
D4F8 CD 1A D6            CALL    SET$
D4FB C3 87 D1            JP      RETXASS
D4FE            ST_SET
D4FE 13                  INC     DE
D4FF 1A                  LD      A,(DE)
D500 FE 41 20            IF A="A" THEN JR START_SET
D503 02 18 4C
D506 CD A2 D5            CALL    SPCCUT
D509                     ;
D509 D5                  PUSH    DE      ;G
D50A 13                  INC     DE      ;R
D50B 13                  INC     DE      ;?
D50C 13                  INC     DE      ;,
D50D 13                  INC     DE      ;?
D50E CD A9 D5            CALL    CCHECK
D511 30 1C               JR      NC,ST_SET2
D513 CD EA D5            CALL    INDEX$
D516 CD 37 D6            CALL    TAB11
D519 11 85 D7            LD      DE,LDE2$
D51C CD 1A D6            CALL    SET$
D51F D1                  POP     DE
D520 01 03 00            LD      BC,3
D523 CD 15 D6            CALL    TRNS$
D526 11 D3 D8            LD      DE,ST$
D529 CD 1A D6            CALL    SET$
D52C C3 87 D1            JP      RETXASS
D52F            ST_SET2
D52F 11 7D D7            LD      DE,LH2$
D532 CD 1A D6            CALL    SET$
D535 01 03 00            LD      BC,3
D538 D1                  POP     DE
D539 CD 15 D6            CALL    TRNS$
D53C D5                  PUSH    DE
D53D 11 F4 D8            LD      DE,ST2$
D540 CD 1A D6            CALL    SET$
D543 D1                  POP     DE
D544 13                  INC     DE      ;,
D545 CD BF D5            CALL    COPY$
D548 2B                  DEC     HL      ;$0D
D549 11 4D D8            LD      DE,LD$2
D54C CD 1A D6            CALL    SET$
D54F C3 87 D1            JP      RETXASS
D552            START_SET
D552 13                  INC     DE      ;R
D553 13                  INC     DE      ;T
D554 CD A2 D5            CALL    SPCCUT
D557 FE 0D               CP      $0D
D559 C8                  RET     Z
D55A D5                  PUSH    DE
D55B 11 34 D8            LD      DE,JMP$
D55E CD 1A D6            CALL    SET$
D561 D1                  POP     DE
D562 CD BF D5            CALL    COPY$
D565 C3 87 D1            JP      RETXASS
D568            SR_SET
D568 13                  INC     DE
D569 1A                  LD      A,(DE)
D56A FE 4C 20            IF A="L" THEN JR SRL_SET
D56D 02 18 09
D570 FE 41 20            IF A="A" THEN JR SRA_SET
D573 02 18 09
D576 C3 7D D1            JP      ERR
D579            SRL_SET
D579 01 1F D9            LD      BC,SRL$
D57C C3 FF D6            JP      SHIFT_SET
D57F            SRA_SET
D57F 01 29 D9            LD      BC,SRA$
D582 C3 FF D6            JP      SHIFT_SET
D585            SL_SET
D585 13                  INC     DE
D586 1A                  LD      A,(DE)
D587 FE 4C 20            IF A="L" THEN JR SLL_SET
D58A 02 18 09
D58D FE 41 20            IF A="A" THEN JR SLA_SET
D590 02 18 09
D593 C3 7D D1            JP      ERR
D596            SLL_SET
D596 01 1A D9            LD      BC,SLL$
D599 C3 FF D6            JP      SHIFT_SET
D59C            SLA_SET
D59C 01 24 D9            LD      BC,SLA$
D59F C3 FF D6            JP      SHIFT_SET
D5A2                     ;
D5A2            SPCCUT
D5A2 13                  INC     DE
D5A3 1A                  LD      A,(DE)
D5A4 FE 20               CP      " "
D5A6 28 FA               JR      Z,SPCCUT
D5A8 C9                  RET
D5A9            CCHECK
D5A9 D5                  PUSH    DE
D5AA            CCHECK2
D5AA 1A                  LD      A,(DE)
D5AB 13                  INC     DE
D5AC FE 0D               CP      $0D
D5AE 28 05               JR      Z,CCHECK3
D5B0 FE 2C               CP      ","
D5B2 20 F6               JR      NZ,CCHECK2
D5B4 37                  SCF
D5B5            CCHECK3
D5B5 D1                  POP     DE
D5B6 C9                  RET
D5B7            LDHL$
D5B7 D5                  PUSH    DE
D5B8 11 76 D7            LD      DE,LH$
D5BB CD 1A D6            CALL    SET$
D5BE D1                  POP     DE
D5BF            COPY$
D5BF 1A                  LD      A,(DE)
D5C0 77                  LD      (HL),A
D5C1 13                  INC     DE
D5C2 23                  INC     HL
D5C3 FE 0D               CP      $0D
D5C5 C8                  RET     Z
D5C6 FE 20               CP      " "
D5C8 28 06               JR      Z,COPY$2
D5CA FE 2C               CP      ","
D5CC 28 02               JR      Z,COPY$2
D5CE 18 EF               JR      COPY$
D5D0            COPY$2
D5D0 3E 0D               LD      A,$0D
D5D2 2B                  DEC     HL
D5D3 77                  LD      (HL),A
D5D4 23                  INC     HL
D5D5 C9                  RET
D5D6            LDHL2$
D5D6 D5                  PUSH    DE
D5D7 11 7D D7            LD      DE,LH2$
D5DA CD 1A D6            CALL    SET$
D5DD D1                  POP     DE
D5DE CD BF D5            CALL    COPY$
D5E1 2B                  DEC     HL
D5E2 3E 29               LD      A,")"
D5E4 77                  LD      (HL),A
D5E5 23                  INC     HL
D5E6 CD 10 D6            CALL    LTNL$
D5E9 C9                  RET
D5EA            INDEX$
D5EA D5                  PUSH    DE
D5EB 11 DB D7            LD      DE,IND$
```


▶ 8月号にあったmicroOdysseyのU氏の文章には感動しました。Oh!X(MZ)がOh!X(MZ)であるための苦労，偉大な人々……。単なる雑誌としてではなく一編の小説のように，いままでのOh!X(MZ)を読み返してしまいました。　原　知久(22)大阪府

```
D5EE CD 1A D6                 CALL    SET$
D5F1 D1                       POP     DE
D5F2               INDEX2
D5F2 1A                       LD      A,(DE)
D5F3 77                       LD      (HL),A
D5F4 13                       INC     DE
D5F5 23                       INC     HL
D5F6 FE 2C                    CP      ","
D5F8 20 F8                    JR      NZ,INDEX2
D5FA 2B                       DEC     HL
D5FB D5                       PUSH    DE
D5FC 11 E2 D7                 LD      DE,IND$2
D5FF CD 1A D6                 CALL    SET$
D602 D1                       POP     DE
D603 01 03 00                 LD      BC,3
D606 CD 15 D6                 CALL    TRNS$
D609 11 EC D7                 LD      DE,IND$3
D60C CD 1A D6                 CALL    SET$
D60F C9                       RET
D610               LTNL$
D610 3E 0D                    LD      A,$0D
D612 77                       LD      (HL),A
D613 23                       INC     HL
D614 C9                       RET
D615               TRNS$
D615 EB                       EX      DE,HL
D616 ED B0                    LDIR
D618 EB                       EX      DE,HL
D619 C9                       RET
D61A               SET$
D61A F5                       PUSH    AF
D61B               SET$2
D61B 1A                       LD      A,(DE)
D61C 77                       LD      (HL),A
D61D FE 0D                    CP      $0D
D61F DC 29 D6                 CALL    C,SET$3
D622 13                       INC     DE
D623 23                       INC     HL
D624 B7                       OR      A
D625 20 F4                    JR      NZ,SET$2
D627 F1                       POP     AF
D628 C9                       RET
D629               SET$3
D629 B7                       OR      A
D62A 28 09                    JR      Z,SET$5
D62C C5                       PUSH    BC
D62D 47                       LD      B,A
D62E 0E 20                    LD      C," "
D630               SET$4
D630 71                       LD      (HL),C
D631 23                       INC     HL
D632 10 FC                    DJNZ    SET$4
D634 C1                       POP     BC
D635               SET$5
D635 2B                       DEC     HL
D636 C9                       RET
D637               TAB11
D637 3E 0C                    LD      A,12
D639 18 EE                    JR      SET$3
D63B               CPAL_SET
D63B C5                       PUSH    BC
D63C 13                       INC     DE          ;AorL
D63D                          ;
D63D CD A2 D5                 CALL    SPCCUT
D640 D5                       PUSH    DE          ;G
D641 13                       INC     DE          ;R
D642 13                       INC     DE          ;?
D643 13                       INC     DE          ;,
D644 13                       INC     DE          ;?
D645 CD A9 D5                 CALL    CCHECK
D648 01 59 D6                 LD      BC,CPAL_ST2
D64B C5                       PUSH    BC
D64C D2 D6 D5                 JP      NC,LDHL2$
D64F C1                       POP     BC
D650 CD EA D5                 CALL    INDEX$
D653 11 5F D8                 LD      DE,LD$3
D656 CD 1A D6                 CALL    SET$
D659               CPAL_ST2
D659 CD 37 D6                 CALL    TAB11
D65C 11 85 D7                 LD      DE,LDE2$
D65F CD 1A D6                 CALL    SET$
D662 D1                       POP     DE
D663 01 03 00                 LD      BC,3
D666 CD 15 D6                 CALL    TRNS$
D669 11 FC D8                 LD      DE,SUB$
D66C CD 1A D6                 CALL    SET$
D66F C1                       POP     BC
D670 50 59                    LD      DE,BC
D672 CD 1A D6                 CALL    SET$
D675 C3 87 D1                 JP      RETXASS
D678               IO_SET
D678 13                       INC     DE          ;NorT
D679                          ;
D679 CD A2 D5                 CALL    SPCCUT
D67C D5                       PUSH    DE
D67D 11 76 D7                 LD      DE,LH$
D680 CD 1A D6                 CALL    SET$
D683 D1                       POP     DE
D684 CD BF D5                 CALL    COPY$
D687 2B                       DEC     HL
D688 D5                       PUSH    DE
D689 11 BD D7                 LD      DE,LDBC$T
D68C CD 1A D6                 CALL    SET$
D68F D1                       POP     DE
D690 CD BF D5                 CALL    COPY$
D693 50 59                    LD      DE,BC
D695 CD 1A D6                 CALL    SET$
D698 C3 87 D1                 JP      RETXASS
D69B               LOG_SET
D69B 13                       INC     DE
D69C                          ;
D69C CD A2 D5                 CALL    SPCCUT
D69F D5                       PUSH    DE          ;G
D6A0 13                       INC     DE          ;R
D6A1 13                       INC     DE          ;?
D6A2 13                       INC     DE          ;,
D6A3 13                       INC     DE          ;?
D6A4 CD A9 D5                 CALL    CCHECK
D6A7 01 B8 D6                 LD      BC,LOG_ST2
D6AA C5                       PUSH    BC
D6AB D2 D6 D5                 JP      NC,LDHL2$
D6AE C1                       POP     BC
D6AF CD EA D5                 CALL    INDEX$
D6B2 11 5F D8                 LD      DE,LD$3
D6B5 CD 1A D6                 CALL    SET$
D6B8               LOG_ST2
D6B8 CD 37 D6                 CALL    TAB11
D6BB 11 85 D7                 LD      DE,LDE2$
D6BE CD 1A D6                 CALL    SET$
D6C1 D1                       POP     DE
D6C2 D5                       PUSH    DE
D6C3 01 03 00                 LD      BC,3
D6C6 CD 15 D6                 CALL    TRNS$
D6C9 11 8D D7                 LD      DE,LOG1$
D6CC CD 1A D6                 CALL    SET$
D6CF ED 5B 46                 LD      DE,(LOG_WRK)
D6D2 D9
D6D3 CD 1A D6                 CALL    SET$
D6D6 11 99 D7                 LD      DE,LOG2$
D6D9 CD 1A D6                 CALL    SET$
D6DC ED 5B 46                 LD      DE,(LOG_WRK)
D6DF D9
D6E0 CD 1A D6                 CALL    SET$
D6E3 11 AD D7                 LD      DE,LOG3$
D6E6 CD 1A D6                 CALL    SET$
D6E9 D1                       POP     DE
D6EA 01 03 00                 LD      BC,3
D6ED CD 15 D6                 CALL    TRNS$
D6F0 11 4D D8                 LD      DE,LD$2
D6F3 CD 1A D6                 CALL    SET$
D6F6 11 53 D8                 LD      DE,CALL_FLAG
D6F9 CD 1A D6                 CALL    SET$
D6FC C3 87 D1                 JP      RETXASS
D6FF               SHIFT_SET
D6FF D5                       PUSH    DE
D700 11 BF D7                 LD      DE,LDBC$
D703 CD 1A D6                 CALL    SET$
D706 50 59                    LD      DE,BC
D708 CD 1A D6                 CALL    SET$
D70B CD 37 D6                 CALL    TAB11
D70E D1                       POP     DE
D70F CD A2 D5                 CALL    SPCCUT
D712 D5                       PUSH    DE          ;G
D713 13                       INC     DE          ;R
D714 13                       INC     DE          ;?
D715 13                       INC     DE          ;,
D716 13                       INC     DE          ;?
D717 CD A9 D5                 CALL    CCHECK
D71A 01 25 D7                 LD      BC,SFT_ST2
D71D C5                       PUSH    BC
D71E D2 B7 D5                 JP      NC,LDHL$
D721 C1                       POP     BC
D722 CD EA D5                 CALL    INDEX$
D725               SFT_ST2
D725 CD 37 D6                 CALL    TAB11
D728 11 85 D7                 LD      DE,LDE2$
D72B CD 1A D6                 CALL    SET$
D72E D1                       POP     DE
D72F D5                       PUSH    DE
D730 01 03 00                 LD      BC,3
D733 CD 15 D6                 CALL    TRNS$
D736 11 C4 D8                 LD      DE,SFT$
D739 CD 1A D6                 CALL    SET$
D73C 11 47 D8                 LD      DE,LD$
D73F CD 1A D6                 CALL    SET$
D742 D1                       POP     DE
D743 01 03 00                 LD      BC,3
D746 CD 15 D6                 CALL    TRNS$
D749 11 4D D8                 LD      DE,LD$2
D74C CD 1A D6                 CALL    SET$
D74F C3 87 D1                 JP      RETXASS
                              ;
                   CPA$
                              DB      11
                              DM      "CALL"
                              DB      4
                              DM      "FLAG_A"
                              DB      $0D
                              DS      1
                   CPL$
                              DB      11
                              DM      "CALL"
                              DB      4
                              DM      "FLAG_L"
                              DB      $0D
                              DS      1
                   DS$
                              DM      "DS"
                              DB      6
                              DS      1
                   DW$
                              DM      "DW"
```

▶景気が悪くなってきたから，と賞与を出し渋っておきながら，あいかわらず休日出勤を強いるいまの会社にメチャメチャ腹が立っています。サマージャンボに賭けるしかないのか？　メモリがほしいよう。

安尾　文教(24)愛知県

```
        DB      6
        DS      1
LH$
        DM      "LD"
        DB      6
        DM      "HL,"
        DS      1
LH2$
        DM      "LD"
        DB      6
        DM      "HL,("
        DS      1
LDE2$
        DM      "LD"
        DB      6
        DM      "DE,("
        DS      1
LOG1$
        DM      ")"
        DB      $0D
        DB      11
        DM      "LD"
        DB      6
        DM      "A,L"
        DB      $0D
        DB      11
        DS      1
LOG2$
        DM      "E"
        DB      $0D
        DB      11
        DM      "LD"
        DB      6
        DM      "L,A"
        DB      $0D
        DB      11
        DM      "LD"
        DB      6
        DM      "A,H"
        DB      $0D
        DB      11
        DS      1
LOG3$
        DM      "D"
        DB      $0D
        DB      11
        DM      "LD"
        DB      6
        DM      "H,A"
        DB      $0D
        DB      11
        DM      "LD"
        DB      6
        DM      "("
        DS      1
LDBC$T
        DB      $0D
        DB      11
LDBC$
        DM      "LD"
        DB      6
        DM      "BC,"
        DS      1
IN$
        DB      11
        DM      "CALL"
        DB      4
        DM      "IN"
        DB      $0D
        DS      1
OUT$
        DB      11
        DM      "CALL"
        DB      4
        DM      "OUT"
        DB      $0D
        DS      1
IND$
        DM      "LD"
        DB      6
        DM      "DE,"
        DS      1
IND$2
        DB      $0D
        DB      11
        DM      "LD"
        DB      6
        DM      "HL,("
        DS      1
IND$3
        DM      ")"
        DB      $0D
        DB      11
        DM      "ADD"
        DB      5
        DM      "HL,DE"
        DB      $0D
        DS      1
JP$
        DM      "LD"
        DB      6
        DM      "A,(FR)"
        DB      $0D
        DB      11
        DM      "CP"
        DB      6
        DM      "$1"
        DS      1
JP_IND$
        DM      ")"
        DB      $0D
        DB      11
        DM      "SLA"
        DB      5
        DM      "L"
        DB      $0D
        DB      11
        DM      "RL"
        DB      6
        DM      "H"
        DB      $0D
        DB      11
        DM      "ADD"
        DB      5
        DM      "HL,DE"
        DB      $0D
        DB      11
        DM      "PUSH"
        DB      4
        DM      "HL"
        DB      $0D
        DB      11
        DM      "RET"
        DB      $0D
        DS      1
JMP$
        DM      "JP"
        DB      6
        DS      1
JPZ$
        DB      11
        DM      "JP"
        DB      6
        DM      "Z,"
        DS      1
JPNZ$
        DB      11
        DM      "JP"
        DB      6
        DM      "NZ,"
        DS      1
LD$
        DB      11
        DM      "LD"
        DB      6
        DM      "("
        DS      1
LD$2
        DM      "),HL"
        DB      $0D
        DS      1
CALL_FLAG
        DB      11
        DM      "CALL"
        DB      4
        DM      "FLAG"
        DB      $0D
        DS      1
LD$3
        DB      11
        DM      "LD"
        DB      6
        DM      "E,(HL)"
        DB      $0D
        DB      11
        DM      "INC"
        DB      5
        DM      "HL"
        DB      $0D
        DB      11
        DM      "LD"
        DB      6
        DM      "D,(HL)"
        DB      $0D
        DB      11
        DM      "EX"
        DB      6
        DM      "DE,HL"
        DB      $0D
        DS      1
PUSH$
        DB      11
        DM      "PUSH"
        DB      4
        DM      "HL"
        DB      $0D
        DB      11
        DM      "LD"
        DB      6
        DM      "(GR4),SP"
        DB      $0D
        DS      1
POP$
        DM      "POP"
        DB      5
        DM      "HL"
        DB      $0D
        DB      11
        DM      "LD"
        DB      6
        DM      "("
        DS      1
```

▶「MATIER」，面白そうですね。楽しみ，楽しみ。　藤原　常雅(21)神奈川県

```
POP$2
              DM      "),HL"
              DB      $0D
              DB      11
              DM      "LD"
              DB      6
              DM      "(GR4),SP"
              DB      $0D
              DS      1
RET$
              DM      "RET"
              DB      $0D
              DS      1
SFT$
              DM      ")"
              DB      $0D
              DB      11
              DM      "CALL"
              DB      4
              DM      "SHIFT"
              DB      $0D
              DS      1
ST$
              DM      ")"
              DB      $0D
              DB      11
              DM      "LD"
              DB      6
              DM      "(HL),E"
              DB      $0D
              DB      11
              DM      "INC"
              DB      5
              DM      "HL"
              DB      $0D
              DB      11
              DM      "LD"
              DB      6
              DM      "(HL),D"
              DB      $0D
              DS      1
ST2$
              DM      ")"
              DB      $0D
              DB      11
              DM      "LD"
              DB      6
              DM      "("
              DS      1
SUB$
              DM      ")"
              DB      $0D
              DB      11
              DM      "EX"
              DB      6
              DM      "DE,HL"
              DB      $0D
              DB      11
              DM      "OR"
              DB      6
              DM      "A"
              DB      $0D
              DB      11
              DM      "SBC"
              DB      5
              DM      "HL,DE"
              DB      $0D
              DS      1
SLL$
              DM      "SLL"
              DB      $0D
              DS      1
SRL$
              DM      "SRL"
              DB      $0D
              DS      1
SLA$
              DM      "SLA"
              DB      $0D
              DS      1
SRA$
              DM      "SRA"
              DB      $0D
              DS      1
TIMES2$
              DM      "*2"
              DB      $0D
              DS      1
AND$
              DM      "AND"
              DB      5
              DS      1
OR$
              DM      "OR"
              DB      6
              DS      1
XOR$
              DM      "XOR"
              DB      5
              DS      1
              ;
EXEADR
              DS      2
TXTADR
              DS      2
WRKADR
              DS      2
LOG_WRK
              DS      2
ORG$
              DM      "       ORG     $"
OYAKUSOKU
              DB      $0D,$0D
              DM      "REG    EQU     $DF00"
              DB      $0D
              DM      "GR0    EQU     REG"
              DB      $0D
              DM      "GR1    EQU     REG+2"
              DB      $0D
              DM      "GR2    EQU     REG+4"
              DB      $0D
              DM      "GR3    EQU     REG+6"
              DB      $0D
              DM      "GR4    EQU     REG+8"
              DB      $0D
              DM      "FR     EQU     REG+10"
              DB      $0D
              DM      "OUT    EQU     REG+11"
              DB      $0D
              DM      "IN     EQU     REG+13"
              DB      $0D
              DM      "SLL    EQU     REG+15"
              DB      $0D
              DM      "SRL    EQU     REG+17"
              DB      $0D
              DM      "SLA    EQU     REG+19"
              DB      $0D
              DM      "SRA    EQU     REG+21"
              DB      $0D
              DM      "SHIFT  EQU     REG+23"
              DB      $0D
              DM      "FLAG_A EQU     REG+25"
              DB      $0D
              DM      "FLAG_L EQU     REG+28"
              DB      $0D
              DM      "FLAG   EQU     REG+30"
              DB      $0D
              DB      $0D
OYA2
KEYBUF                ;

DF00                          ORG     PRE+$0F00
DF00                 GR0
DF00 00 00                    DS      2
DF02                 GR1
DF02 00 00                    DS      2
DF04                 GR2
DF04 00 00                    DS      2
DF06                 GR3
DF06 00 00                    DS      2
DF08                 GR4
DF08 00 00                    DS      2
DF0A                 FR
DF0A 00                       DS      1
DF0B                 OUT
DF0B 18 27                    JR      OUT_EXE
DF0D                 IN
DF0D 18 36                    JR      IN_EXE
DF0F                 SLL
DF0F 18 5C                    JR      SLL_EXE
DF11                 SRL
DF11 18 5F                    JR      SRL_EXE
DF13                 SLA
DF13 18 62                    JR      SLA_EXE
DF15                 SRA
DF15 18 69                    JR      SRA_EXE
DF17                 SHIFT
DF17 18 74                    JR      SHIFT_EXE
DF19
DF19                 ;        FLAG CHECK input = F
DF19
DF19                 FLAG_A
DF19 C3 A2 DF                 JP      FL_A
DF1C                 FLAG_L
DF1C CB 1C                    RR      H
DF1E
DF1E                 ;        FLAG CHECK input = HL
DF1E
DF1E                 FLAG
DF1E F5                       PUSH    AF
DF1F 7C                       LD      A,H
DF20 B5                       OR      L
DF21 20 03                    JR      NZ,FLAG2
DF23 3C                       INC     A       ; zero = 01
DF24 18 07                    JR      FLAG3
DF26                 FLAG2
DF26 AF                       XOR     A
DF27 CB 04                    RLC     H
DF29 1F                       RRA,
DF2A 1F                       RRA
DF2B 1F                       RRA     ; plus  = 00
DF2C 1F                       RRA     ; minus = 10
DF2D                 FLAG3
DF2D 32 0A DF                 LD      (FR),A
DF30 CB 0C                    RRC     H       ; POP HL
DF32 F1                       POP     AF
DF33 C9                       RET
DF34                 OUT_EXE
DF34 0A                       LD      A,(BC)
DF35 5F                       LD      E,A
DF36 03                       INC     BC
DF37 0A                       LD      A,(BC)
```

▶現在，群馬県にいます。この半年，横浜（鶴見），千葉，横浜（港北），大阪，横浜，群馬と１カ月くらいごとに移り住んでいます。８月にはやっと落ち着くとは思いますが，明日にはどこへ行くのかわからない，というサバイバルな毎日でした。やっぱり大企業ってヘン。　　奥田　健児(22)千葉県

```
DF38 57                      LD      D,A
DF39             OUT_EXE2
DF39 7E                      LD      A,(HL)
DF3A CD F4 1F                CALL    #PRINT
DF3D 23                      INC     HL        ;0
DF3E 23                      INC     HL        ;CHR
DF3F 1B                      DEC     DE
DF40 7B                      LD      A,E
DF41 B2                      OR      D
DF42 20 F5                   JR      NZ,OUT_EXE2
DF44 C9                      RET
DF45             IN_EXE
DF45 54 5D                   LD      DE,HL
DF47 CD D3 1F                CALL    #GETL
DF4A 2B                      DEC     HL
DF4B             IN_EXE2
DF4B 23                      INC     HL
DF4C 7E                      LD      A,(HL)
DF4D B7                      OR      A
DF4E 20 FB                   JR      NZ,IN_EXE2
DF50                         ;
DF50 B7                      OR      A
DF51 ED 52                   SBC     HL,DE
DF53 7D                      LD      A,L
DF54 02                      LD      (BC),A
DF55 03                      INC     BC
DF56 AF                      XOR     A
DF57 02                      LD      (BC),A
DF58 47                      LD      B,A       ;B=0
DF59 7D                      LD      A,L       ;A=Count
DF5A B7                      OR      A
DF5B C8                      RET     Z
DF5C             IN_EXE3
DF5C F5                      PUSH    AF
DF5D 4F                      LD      C,A
DF5E 0D                      DEC     C
DF5F 62 6B                   LD      HL,DE     ;HL=WORD_ADR
DF61 09                      ADD     HL,BC
DF62 7E                      LD      A,(HL)
DF63 09                      ADD     HL,BC
DF64 77                      LD      (HL),A
DF65 23                      INC     HL
DF66 AF                      XOR     A
DF67 77                      LD      (HL),A
DF68 F1                      POP     AF
DF69 3D                      DEC     A
DF6A 20 F0                   JR      NZ,IN_EXE3
DF6C C9                      RET
DF6D             SLL_EXE
DF6D CB 23                   SLA     E
DF6F CB 12                   RL      D
DF71 C9                      RET
DF72             SRL_EXE
DF72 CB 3A                   SRL     D
DF74 CB 1B                   RR      E
DF76 C9                      RET
DF77             SLA_EXE
DF77 CB 23                   SLA     E
DF79 CB 12                   RL      D
DF7B CB 12                   RL      D
DF7D CB 0A                   RRC     D
DF7F C9                      RET
DF80             SRA_EXE
DF80 CB 3A                   SRL     D
DF82 CB 1B                   RR      E
DF84 CB 02                   RLC     D
DF86 CB 02                   RLC     D
DF88 CB 1A                   RR      D
DF8A CB 1A                   RR      D
DF8C C9                      RET
DF8D             SHIFT_EXE
DF8D 7D                      LD      A,L
DF8E B4                      OR      H
DF8F 28 0C                   JR      Z,SHIFT3;count=0
DF91                         ;
DF91 C5                      PUSH    BC        ;EX BC,HL
DF92 E5                      PUSH    HL
DF93 C1                      POP     BC
DF94 E1                      POP     HL
DF95             SHIFT2
DF95 CD 81 1F                CALL    [HL]      ;S-OS
DF98 0B                      DEC     BC        ;BC=count
DF99 79                      LD      A,C       ;DE=(GR?)
DF9A B0                      OR      B         ;HL=Shift
DF9B 20 F8                   JR      NZ,SHIFT2
DF9D             SHIFT3
DF9D EB                      EX      DE,HL
DF9E CD 1E DF                CALL    FLAG
DFA1 C9                      RET
DFA2             FL_A
DFA2 01 1C DF                LD      BC,FLAG_L
DFA5 C5                      PUSH    BC
DFA6 30 13                   JR      NC,FL_PLUS
DFA8 FA AD DF                JP      M,FL_MINUS
DFAB 3F                      CCF
DFAC C9                      RET               ;S=0,C=1
DFAD             FL_MINUS
DFAD E5                      PUSH    HL
DFAE 11 00 80                LD      DE,$8000
DFB1 B7                      OR      A
DFB2 ED 52                   SBC     HL,DE
DFB4 E1                      POP     HL
DFB5 C2 B9 DF                JP      NZ,FL_MIN2
DFB8 C9                      RET               ;S=1,C=1,HL=$8000
DFB9             FL_MIN2
DFB9 37                      SCF
DFBA C9                      RET               ;S=1,C=1,HL<>$8000
DFBB             FL_PLUS
DFBB F0                      RET     P   ;S=0,C=0
DFBC 3F                      CCF
DFBD C9                      RET               ;S=1,C=0
OBJECT CODE END DFBD
```

リスト5

```
D000 CD E2 1F 54 45 58 54 20 : 33
D008 41 64 64 72 65 73 73 20 : E6
D010 3D 20 24 00 CD 37 D0 22 : 77
D018 42 D9 CD E2 1F 45 58 45 : CB
D020 43 20 41 64 64 72 65 73 : B6
D028 73 20 3D 20 24 00 CD 37 : 18
D030 D0 22 40 D9 C3 6F D0 11 : 1E
D038 DB DA CD D3 1F 1A FE 1B : A7
D040 CA 5E D0 11 EB DA CD B2 : 4D
D048 1F D0 CD E2 1F 53 79 6E : F7
D050 74 61 78 20 45 72 72 6F : 05
D058 72 2E 00 CD EE 1F 2A 42 : E6
D060 D9 AF 77 CD E2 1F 45 4E : 60
D068 44 2E 00 CD EE 1F C9 2A : 3F
D070 42 D9 11 00 00 ED 53 44 : B0
D078 D9 7E EB CD 9A 1F EB 13 : C6
--------------------------------
SUM: F5 6C 87 1F A7 4A 1D 1D F123

D080 23 B7 20 F5 ED 5B 42 D9 : 52
D088 21 48 D9 01 11 00 ED B0 : F1
D090 EB 3A 41 D9 CD AC D0 3A : C2
D098 40 D9 CD AC D0 EB 21 59 : C7
D0A0 D9 01 82 01 ED B0 ED 53 : 3A
D0A8 42 D9 18 11 F5 0F 0F 0F : 66
D0B0 0F CD BB 1F 77 23 F1 CD : 0E
D0B8 BB 1F 77 23 C9 11 DB DA : 03
D0C0 2A 44 D9 CD 94 1F B7 28 : A6
D0C8 95 12 13 23 FE 0D 28 02 : 12
D0D0 18 F1 22 44 D9 11 DB DA : 0E
D0D8 CD E8 1F CD EE 1F CD E8 : 63
D0E0 D0 CD C7 1F 5E D0 18 D5 : 9E
D0E8 2A 42 D9 3E 3B 77 23 1A : 72
D0F0 77 13 23 FE 0D 20 F8 22 : F2
D0F8 42 D9 11 DB DA 1A FE 20 : 19
--------------------------------
SUM: AB 02 D4 06 96 C2 A0 42 6640

D100 28 18 FE 0D 28 14 FE 3B : C0
D108 28 7D 1A 77 23 13 FE 0D : 77
D110 28 75 FE 20 20 F4 CD 10 : AC
D118 D6 AF FE 0D 28 69 1A FE : 39
D120 20 13 28 F6 FE 3B 20 03 : AD
D128 C3 87 D1 F5 CD 37 D6 F1 : DB
D130 FE 41 20 03 C3 8B D1 FE : 7F
D138 43 20 03 C3 F5 D1 FE 44 : 31
D140 20 03 C3 29 D2 FE 45 20 : 44
D148 03 C3 AA D2 FE 49 20 03 : AC
D150 C3 D5 D2 FE 4A 20 03 C3 : 98
D158 DB D2 FE 4C 20 03 C3 AE : 8B
D160 D3 FE 4F 20 03 C3 20 D4 : FA
D168 FE 50 20 03 C3 40 D4 FE : 46
D170 52 20 03 C3 87 D4 FE 53 : E4
D178 20 03 C3 90 D4 CD C4 1F : FA
--------------------------------
SUM: 76 92 A2 1D 71 60 89 64 6D40

D180 3E 3F 77 23 CD 10 D6 22 : EC
D188 42 D9 C9 1A FE 44 20 02 : 62
D190 18 13 FE 4E 20 02 18 03 : B4
D198 C3 7D D1 01 32 D9 ED 43 : 4D
D1A0 46 D9 C3 9B D6 13 CD A2 : D5
D1A8 D5 D5 13 13 13 13 CD A9 : 6C
D1B0 D5 01 C2 D1 C5 D2 D6 D5 : AB
D1B8 C1 CD EA D5 11 5F D8 CD : 62
D1C0 1A D6 CD 37 D6 11 85 D7 : 37
D1C8 CD 1A D6 D1 D5 01 03 00 : 67
D1D0 CD 15 D6 11 EC D7 CD 1A : 73
D1D8 D6 11 47 D8 CD 1A D6 D1 : 94
D1E0 01 03 00 CD 15 D6 11 4D : 1A
D1E8 D8 CD 1A D6 11 53 D8 CD : 9E
D1F0 1A D6 C3 87 D1 1A FE 41 : 64
D1F8 20 02 18 09 FE 50 20 02 : B3
--------------------------------
SUM: A9 E2 46 04 35 1C 75 76 BBC6

D200 18 0A C3 7D D1 1B CD BF : DA
D208 D5 C3 87 D1 13 1A FE 4C : 67
D210 20 02 18 09 FE 41 20 02 : A4
D218 18 09 C3 7D D1 01 60 D7 : 6A
D220 C3 3B D6 01 52 D7 C3 3B : FC
D228 D6 1A FE 43 20 02 18 09 : 74
D230 FE 53 20 02 18 5C C3 7D : 27
D238 D1 CD A2 D5 D5 1A FE 23 : 25
D240 20 02 18 13 FE 22 20 02 : 8F
D248 18 1F 11 72 D7 CD 1A D6 : 4E
D250 D1 CD BF D5 C3 87 D1 11 : 5E
D258 72 D7 CD 1A D6 D1 3E 24 : 39
D260 77 13 23 CD BF D5 C3 87 : 58
D268 D1 11 0B 00 B7 ED 52 D1 : B4
D270 13 1A FE 22 CA 87 D1 D5 : 44
D278 F5 CD 37 D6 11 72 D7 CD : F6
--------------------------------
SUM: 58 1D D3 28 D1 C8 ED CF 2D17

D280 1A D6 3E 22 77 23 F1 77 : 52
D288 23 3E 22 77 23 CD 10 D6 : D0
D290 18 DD CD A2 D5 D5 11 6E : 8D
D298 D7 CD 1A D6 D1 CD BF D5 : C6
D2A0 2B 11 2E D9 CD 1A D6 C3 : C3
D2A8 87 D1 1A FE 4E 20 03 C3 : A4
D2B0 C2 D2 FE 4F 20 02 18 13 : 2E
D2B8 FE 58 20 03 C3 C2 D2 C3 : 93
D2C0 7D D1 11 BF D8 CD 1A D6 : B3
D2C8 C3 87 D1 01 3B D9 ED 43 : 60
D2D0 46 D9 C3 9B D6 01 C6 D7 : F1
D2D8 C3 78 D6 1A FE 5A 20 02 : A5
D2E0 18 39 FE 4E 20 02 18 4E : 25
D2E8 FE 50 20 02 18 63 FE 4D : 36
D2F0 28 03 C3 7D D1 13 1A FE : 67
D2F8 50 20 02 18 73 CD A2 D5 : 41
--------------------------------
SUM: 75 1F 0B 94 A1 D6 53 4C 289D

D300 D5 11 FA D7 CD 1A D6 3E : B2
D308 30 77 23 CD 10 D6 11 38 : C6
D310 D8 CD 1A D6 D1 CD BF D5 : C7
D318 C3 87 D1 13 CD A2 D5 D5 : 47
D320 11 FA D7 CD 1A D6 CD 10 : 7C
D328 D6 11 38 D8 CD 1A D6 D1 : 85
D330 CD BF D5 C3 87 D1 13 CD : 5C
D338 A2 D5 D5 11 FA D7 CD 1A : 15
D340 D6 CD 10 D6 11 3F D8 CD : 7E
D348 1A D6 D1 CD BF D5 C3 87 : 6C
D350 D1 13 CD A2 D5 D5 11 FA : 08
D358 D7 CD 1A D6 3E 30 77 23 : 9C
```


▶ついに，SX-WINDOWの開発キットが発売される模様ですね。広告によると9月発売予定となっていますね。ということは12月だな。　大久保　明弘(20)岩手県

```
D360 CD 10 D6 11 3F D8 CD 1A : C2
D368 D6 D1 CD BF D5 C3 87 D1 : 23
D370 CD A2 D5 CD A9 D5 38 0E : D5
D378 D5 11 34 D8 CD 1A D6 D1 : 80
---------------------------------
SUM: D3 92 35 96 50 9A 83 23 0D3D

D380 CD BF D5 C3 87 D1 D5 11 : 62
D388 DB D7 CD 1A D6 D1 1A 77 : D1
D390 13 23 FE 2C 20 F8 2B D5 : 78
D398 11 E2 D7 CD 1A D6 D1 01 : 59
D3A0 03 00 CD 15 D6 11 0B D8 : AF
D3A8 CD 1A D6 C3 87 D1 1A FE : F0
D3B0 44 20 02 18 09 FE 45 20 : EA
D3B8 02 18 35 C3 7D D1 CD A2 : CF
D3C0 D5 D5 13 13 13 13 CD A9 : 6C
D3C8 D5 01 DA D3 C5 D2 D6 D5 : C5
D3D0 C1 CD EA D5 11 5F D8 CD : 62
D3D8 1A D6 11 47 D8 CD 1A D6 : DD
D3E0 D1 01 03 00 CD 15 D6 11 : 9E
D3E8 4D D8 CD 1A D6 C3 87 D1 : FD
D3F0 13 CD A2 D5 D5 13 13 13 : 65
D3F8 13 CD A9 D5 01 07 D4 C5 : FF
---------------------------------
SUM: AB D9 54 4F B4 24 FB D1 2CAC

D400 D2 B7 D5 C1 CD EA D5 11 : BC
D408 47 D8 CD 1A D6 D1 01 03 : B1
D410 00 CD 15 D6 11 4D D8 CD : BB
D418 1A D6 CD 1A D6 C3 87 D1 : C8
D420 1A FE 52 20 02 18 09 FE : AB
D428 55 20 02 18 0D C3 7D D1 : AD
D430 01 37 D9 ED 43 46 D9 C3 : 23
D438 9B D6 01 D0 D7 C3 78 D6 : 2A
D440 1A FE 55 20 02 18 09 FE : AE
D448 4F 20 02 18 1F C3 7D D1 : B9
D450 13 13 CD A2 D5 CD A9 D5 : B5
D458 01 63 D4 C5 D2 B7 D5 C1 : 1C
D460 CD EA D5 11 88 D8 CD 1A : E4
D468 D6 C3 87 D1 D5 11 9F D8 : 4E
D470 CD 1A D6 D1 13 CD A2 D5 : E5
D478 01 03 00 CD 15 D6 11 AC : 79
---------------------------------
SUM: 2C BB DC DF 00 9A 2F F2 AD78

D480 D8 CD 1A D6 C3 87 D1 11 : C1
D488 BF D8 CD 1A D6 C3 87 D1 : 6F
D490 1A FE 55 20 02 18 17 FE : BC
D498 54 20 02 18 61 FE 52 20 : 5F
D4A0 03 C3 68 D5 FE 4C 20 03 : 70
D4A8 C3 85 D5 C3 7D D1 13 CD : 0E
D4B0 A2 D5 D5 13 13 13 13 CD : 65
D4B8 A9 D5 01 CB D4 C5 D2 D6 : 8B
D4C0 D5 C1 CD EA D5 11 5F D8 : 6A
D4C8 CD 1A D6 CD 37 D6 11 85 : 2D
D4D0 D7 CD 1A D6 D1 D5 01 03 : 3E
D4D8 00 CD 15 D6 11 FC D8 CD : 6A
D4E0 1A D6 11 47 D8 CD 1A D6 : DD
D4E8 D1 01 03 00 CD 15 D6 11 : 9E
D4F0 4D D8 CD 1A D6 11 53 D8 : 1E
D4F8 CD 1A D6 C3 87 D1 13 1A : 05
---------------------------------
SUM: 94 F3 DA 25 4E D1 78 79 B735

D500 FE 41 20 02 18 4C CD A2 : 34
D508 D5 D5 13 13 13 13 CD A9 : 6C
D510 D5 30 1C CD EA D5 CD 37 : B1
D518 D6 11 85 D7 CD 1A D6 D1 : D1
D520 01 03 00 CD 15 D6 11 D3 : A0
D528 D8 CD 1A D6 C3 87 D1 11 : C1
D530 7D D7 CD 1A D6 01 03 00 : 15
D538 D1 CD 15 D6 D5 11 F4 D8 : 3B
D540 CD 1A D6 D1 13 CD BF D5 : 02
D548 2B 11 4D D8 CD 1A D6 C3 : E1
D550 87 D1 13 13 CD A2 D5 FE : C0
D558 0D C8 D5 11 34 D8 CD 1A : AE
D560 D6 D1 CD BF D5 C3 87 D1 : 23
D568 13 1A FE 4C 20 02 18 09 : BA
D570 FE 41 20 02 18 09 C3 7D : C2
D578 D1 01 1F D9 C3 FF D6 01 : 63
---------------------------------
SUM: E9 BC E5 FF 16 EB 85 17 7052

D580 29 D9 C3 FF D6 13 1A FE : C5
D588 4C 20 02 18 09 FE 41 20 : EE
D590 02 18 09 C3 7D D1 01 1A : 4F
D598 D9 C3 FF D6 01 24 D9 C3 : 32
D5A0 FF D6 13 1A FE 20 28 FA : 42
D5A8 C9 D5 1A 13 FE 0D 28 05 : 03
D5B0 FE 2C 20 F6 37 D1 C9 D5 : E6
D5B8 11 76 D7 CD 1A D6 D1 1A : 06
D5C0 77 13 23 FE 0D C8 FE 20 : 9E
D5C8 28 06 FE 2C 28 02 18 EF : 89
D5D0 3E 0D 2B 77 23 C9 D5 11 : BF
D5D8 7D D7 CD 1A D6 D1 CD BF : 6E
D5E0 D5 2B 3E 29 77 23 CD 10 : DE
D5E8 D6 C9 D5 11 DB D7 CD 1A : 1E
D5F0 D6 D1 1A 77 13 23 FE 2C : 98
D5F8 20 F8 2B D5 11 E2 D7 CD : AF
---------------------------------
SUM: 22 DB 62 E1 4E 3D 46 EB 00F9

D600 1A D6 D1 01 03 00 CD 15 : A7
D608 D6 11 EC D7 CD 1A D6 C9 : 30
D610 3E 0D 77 23 C9 EB ED B0 : 36
D618 EB C9 F5 1A 77 FE 0D DC : 21
D620 29 D6 13 23 B7 20 F4 F1 : F1
D628 C9 B7 28 09 C5 47 0E 20 : EB
D630 71 23 10 FC C1 2B C9 3E : 93
D638 0C 18 EE C5 13 CD A2 D5 : 2E
D640 D5 13 13 13 13 CD A9 D5 : 6C
D648 01 59 D6 C5 D2 D6 D5 C1 : 33
D650 CD EA D5 11 5F D8 CD 1A : BB
D658 D6 CD 37 D6 11 85 D7 CD : EA
D660 1A D6 D1 01 03 00 CD 15 : A7
D668 D6 11 FC D8 CD 1A D6 C1 : 39
D670 50 59 CD 1A D6 C3 87 D1 : 81
D678 13 CD A2 D5 D5 11 76 D7 : 8A
---------------------------------
SUM: 54 B5 93 89 30 50 CC 89 AB3C

D680 CD 1A D6 D1 CD BF D5 2B : 1A
D688 D5 11 BD D7 CD 1A D6 D1 : 08
D690 CD BF D5 50 59 CD 1A D6 : C7
D698 C3 87 D1 13 CD A2 D5 D5 : 47
D6A0 13 13 13 13 CD A9 D5 01 : 98
D6A8 B8 D6 C5 D2 D6 D5 C1 CD : 5E
D6B0 EA D5 11 5F D8 CD 1A D6 : C4
D6B8 CD 37 D6 11 85 D7 CD 1A : 2E
D6C0 D6 D1 D5 01 03 00 CD 15 : 62
D6C8 D6 11 8D D7 CD 1A D6 ED : F5
D6D0 5B 46 D9 CD 1A D6 11 99 : E1
D6D8 D7 CD 1A D6 ED 5B 46 D9 : FB
D6E0 CD 1A D6 11 AD D7 CD 1A : 39
D6E8 D6 D1 01 03 00 CD 15 D6 : 63
D6F0 11 4D D8 CD 1A D6 11 53 : 57
D6F8 D8 CD 1A D6 C3 87 D1 D5 : 85
---------------------------------
SUM: 1E 60 16 92 21 B6 D5 F1 D202

D700 11 BF D7 CD 1A D6 50 59 : 0D
D708 CD 1A D6 CD 37 D6 D1 CD : 35
D710 A2 D5 D5 13 13 13 13 CD : 65
D718 A9 D5 01 25 D7 C5 D2 B7 : C9
D720 D5 C1 CD EA D5 CD 37 D6 : FC
D728 11 85 D7 CD 1A D6 D1 D5 : D0
D730 01 03 00 CD 15 D6 11 C4 : 91
D738 D8 CD 1A D6 11 47 D8 CD : 92
D740 1A D6 D1 01 03 00 CD 15 : A7
D748 D6 11 4D D8 CD 1A D6 C3 : 8C
D750 87 D1 0B 43 41 4C 4C 04 : 83
D758 46 4C 41 47 5F 41 0D 00 : C7
D760 0B 43 41 4C 4C 04 46 4C : BD
D768 41 47 5F 4C 0D 00 44 53 : D7
D770 06 00 44 57 06 00 4C 44 : 37
D778 06 48 4C 2C 00 4C 44 06 : 5C
---------------------------------
SUM: FD 6F DB AA 1F 3B 0D AB 9D33

D780 48 4C 2C 28 00 4C 44 06 : 7E
D788 44 45 2C 28 00 29 0D 0B : 1E
D790 4C 44 06 41 2C 4C 0D 0B : 67
D798 00 45 0D 0B 4C 44 06 4C : 3F
D7A0 2C 41 0D 0B 4C 44 06 41 : 5C
D7A8 2C 48 0D 0B 00 44 0D 0B : E8
D7B0 4C 44 06 48 2C 41 0D 0B : 63
D7B8 4C 44 06 28 00 0D 0B 4C : 22
D7C0 44 06 42 43 2C 00 0B 43 : 49
D7C8 41 4C 4C 04 49 4E 0D 00 : 81
D7D0 0B 43 41 4C 4C 04 4F 55 : CF
D7D8 54 0D 00 4C 44 06 44 45 : 80
D7E0 2C 00 0D 0B 4C 44 06 48 : 22
D7E8 4C 2C 28 00 29 0D 0B 41 : 22
D7F0 44 44 05 48 4C 2C 44 45 : D6
D7F8 0D 00 4C 44 06 41 2C 28 : 38
---------------------------------
SUM: 75 3D E6 98 BC F1 BB DE B1F4

D800 46 52 29 0D 0B 43 50 06 : 72
D808 24 31 00 29 0D 0B 53 4C : 35
D810 41 05 4C 0D 0B 52 4C 06 : 4E
D818 48 0D 0B 41 44 44 05 48 : 76
D820 4C 2C 44 45 0D 0B 50 55 : BE
D828 53 48 04 48 4C 0D 0B 52 : 9D
D830 45 54 0D 00 4A 50 06 00 : 46
D838 0B 4A 50 06 5A 2C 00 0B : 3C
D840 4A 50 06 4E 5A 2C 00 0B : 7F
D848 4C 44 06 28 00 29 2C 48 : 5B
D850 4C 0D 00 0B 43 41 4C 4C : 80
D858 04 46 4C 41 47 0D 00 0B : 36
D860 4C 44 06 45 2C 28 48 4C : C3
D868 29 0D 0B 49 4E 43 05 48 : 68
D870 4C 0D 0B 4C 44 06 44 2C : 6A
D878 28 48 4C 29 0D 0B 45 58 : 9A
---------------------------------
SUM: B1 34 E5 DC 13 97 A3 14 F706

D880 06 44 45 2C 48 4C 0D 00 : 5C
D888 0B 50 55 53 48 04 48 4C : E3
D890 0D 0B 4C 44 06 28 47 52 : 6F
D898 34 29 2C 53 50 0D 00 50 : 89
D8A0 4F 50 05 48 4C 0D 0B 4C : 9C
D8A8 44 06 28 00 29 2C 48 4C : 5B
D8B0 0D 0B 4C 44 06 28 47 52 : 6F
D8B8 34 29 2C 53 50 0D 00 52 : 8B
D8C0 45 54 0D 00 29 0D 0B 43 : 2A
D8C8 41 4C 4C 04 53 48 49 46 : 07
D8D0 54 0D 00 29 0D 0B 4C 44 : 32
D8D8 06 28 48 4C 29 2C 45 0D : 69
D8E0 0B 49 4E 43 05 48 4C 0D : 8B
D8E8 0B 4C 44 06 28 48 4C 29 : 86
D8F0 2C 44 0D 00 29 0D 0B 4C : 0A
D8F8 44 06 28 00 29 0D 0B 45 : F8
---------------------------------
SUM: 8C 06 1F B7 E2 29 C9 CB 4601

D900 58 06 44 45 2C 48 4C 0D : B4
D908 0B 4F 52 06 41 0D 0B 53 : 5E
D910 42 43 05 48 4C 2C 44 45 : D3
D918 0D 00 53 4C 4C 0D 00 53 : 58
D920 52 4C 0D 00 53 4C 41 0D : 98
D928 00 53 52 41 0D 00 2A 32 : 4F
D930 0D 00 41 4E 44 05 00 4F : 34
D938 52 06 00 58 4F 52 05 00 : 56
D940 00 00 00 00 00 00 00 00 : 00
D948 20 20 20 20 20 20 20 20 : 00
D950 4F 52 47 20 20 20 20 20 : 88
D958 24 0D 0D 52 45 47 20 20 : 5C
D960 20 20 20 45 51 55 20 20 : 8B
D968 20 20 20 24 44 46 30 30 : 6E
D970 0D 47 52 30 20 20 20 20 : 56
D978 20 45 51 55 20 20 20 20 : 8B
---------------------------------
SUM: 63 88 E5 46 52 93 FB 76 E36B

D980 20 52 45 47 0D 47 52 31 : D5
D988 20 20 20 20 20 45 51 55 : 8B
D990 20 20 20 20 20 52 45 47 : 7E
D998 2B 32 0D 47 52 32 20 20 : 75
D9A0 20 20 20 45 51 55 20 20 : 8B
D9A8 20 20 20 52 45 47 2B 34 : 9D
D9B0 0D 47 52 33 20 20 20 20 : 59
D9B8 20 45 51 55 20 20 20 20 : 8B
D9C0 20 52 45 47 2B 36 0D 47 : B3
D9C8 52 34 20 20 20 20 20 45 : 6B
D9D0 51 55 20 20 20 20 20 52 : 98
D9D8 45 47 2B 38 0D 46 52 20 : B4
D9E0 20 20 20 20 20 45 51 55 : 8B
D9E8 20 20 20 20 20 52 45 47 : 7E
D9F0 2B 31 30 0D 4F 55 54 20 : B1
D9F8 20 20 20 20 45 51 55 20 : 8B
---------------------------------
SUM: 8B 43 B5 19 C1 E5 71 5B 4D20

DA00 20 20 20 20 52 45 47 2B : 89
DA08 31 31 0D 49 4E 20 20 20 : 66
DA10 20 20 20 45 51 55 20 20 : 8B
DA18 20 20 20 52 45 47 2B 31 : 9A
DA20 33 0D 53 4C 4C 20 20 20 : 8B
DA28 20 20 45 51 55 20 20 20 : 8B
DA30 20 20 52 45 47 2B 31 35 : AF
DA38 0D 53 52 4C 20 20 20 20 : 7E
DA40 20 45 51 55 20 20 20 20 : 8B
DA48 20 52 45 47 2B 31 37 0D : 9E
DA50 53 4C 41 20 20 20 20 20 : 80
DA58 45 51 55 20 20 20 20 20 : 8B
DA60 52 45 47 2B 31 39 0D 53 : D3
DA68 52 41 20 20 20 20 20 45 : 78
DA70 51 55 20 20 20 20 20 52 : 98
DA78 45 47 2B 32 31 0D 53 48 : C2
---------------------------------
SUM: 23 87 87 A7 6B A3 7A D0 6D66

DA80 49 46 54 20 20 20 45 51 : D9
DA88 55 20 20 20 20 20 52 45 : 8C
DA90 47 2B 32 33 0D 46 4C 41 : B7
DA98 47 5F 41 20 20 45 51 55 : 12
DAA0 20 20 20 20 20 52 45 47 : 7E
DAA8 2B 32 35 0D 46 4C 41 47 : B9
DAB0 5F 4C 20 20 45 51 55 20 : F6
DAB8 20 20 20 20 52 45 47 2B : 89
DAC0 32 38 0D 46 4C 41 47 20 : B1
DAC8 20 20 20 45 51 55 20 20 : 8B
DAD0 20 20 20 52 45 47 2B 33 : 9C
DAD8 30 0D 0D                : 4A
---------------------------------
SUM: 98 33 D6 DD 4C DC E8 78 7282

DF00 00 00 00 00 00 00 00 00 : 00
DF08 00 00 00 18 27 18 36 18 : A5
DF10 5C 18 5F 18 62 18 69 18 : E6
DF18 74 C3 A2 DF CB 1C F5 7C : 10
DF20 B5 20 03 3C 18 07 AF CB : AD
DF28 04 1F 1F 1F 1F 32 0A DF : 9B
DF30 CB 0C F1 C9 0A 5F 03 0A : 07
DF38 57 7E CD F4 1F 23 23 1B : 16
DF40 7B B2 20 F5 C9 54 5D CD : 89
DF48 D3 1F 2B 23 7E B7 20 FB : 90
DF50 B7 ED 52 7D 02 03 AF 02 : 29
DF58 47 7D B7 C8 F5 4F 0D 62 : F6
DF60 6B 09 7E 09 77 23 AF 77 : BB
DF68 F1 3D 20 F0 C9 CB 23 CB : C0
DF70 12 C9 CB 3A CB 1B C9 CB : 5A
DF78 23 CB 12 CB 12 CB 0A C9 : 7B
---------------------------------
SUM: 88 B9 B0 82 0F 38 51 7D B6D0

DF80 CB 3A CB 1B CB 02 CB 02 : 85
DF88 CB 1A CB 1A C9 7D B4 28 : EC
DF90 0C C5 E5 C1 E1 CD 81 1F : C5
DF98 0B 79 B0 20 F8 EB CD 1E : 22
DFA0 DF C9 01 1C DF C5 30 13 : AC
DFA8 FA AD DF 3F C9 E5 11 00 : 84
DFB0 80 B7 ED 52 E1 C2 B9 DF : B1
DFB8 C9 37 C9 F0 3F C9       : C1
---------------------------------
SUM: CF F6 C1 B3 35 6C C7 59 A31E
```

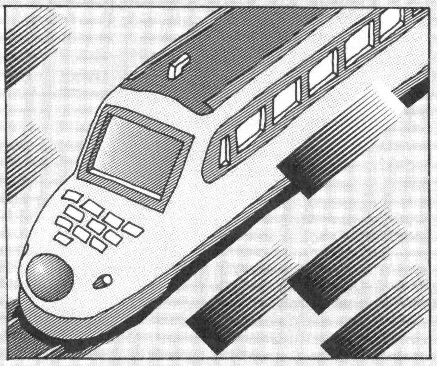
▶最近になって２進数の加算，減算ができるようになりました。実は，１の補数とか２の補数とか使うのですね。パソコン歴４年を経てこんなことをいっているのは，なんだか恥ずかしい。

吉田　勝昌(19)群馬県

X-OVER・NIGHT
(クロスオーバーナイト)

[第26話]

新幹線とコンピュータ



TAKAHARA HIDEKI 高原 秀己

昔むかし，その昔。東京オリンピックに合わせて，1964年に「夢の超特急」として開業した新幹線。いうまでもなく，いまだに日本では最も速い旅客列車だ。安定してあれだけの本数を運転している点まで考慮すれば，スピードはともかく，世界的に見ても最も優れた列車といっていいかもしれない。

「夢の超特急」と呼ばれていた背景には，デザインのよさもあったはず。とくに“顔”の部分のデザインは抜群だ。つい最近までほかの列車のデザインがあまりにもダサすぎたこともあって，30年たったいまでも「夢の超特急」の称号はおかしくない。そういえば，開業の少しあとにオンエアされた「ウルトラマン」とは奇妙なことに，“顔”のデザインが似ているなあ（と思うのだが賛否両論あるかもしれない）。

さて，新幹線が誕生して20数年。東海道新幹線に山陽新幹線部分が延長されて，さらに東北新幹線，上越新幹線の２線が加わってきたが，ハード的にはほぼオリジナルを踏襲してきた。しかし，ウルトラマンではないが，新幹線にもこのところやたらと仲間が増えてきている。東海道スーパー新幹線の「のぞみ」に続いて，７月１日には山形新幹線「つばさ」がデビューした。ただ，この弟分である山形新幹線，ちょっとこれまでの兄貴分の新幹線たちとは様子が違っている。

いざ開業してみると，故障やら整備不良とかで，７月の間，連日のように問題を起こしていた。新幹線は開業以来一度も人身事故を起こしていないのがご自慢だが，今回初めて発生してしまうのではないか，とマスコミは固唾を飲んで待機する始末（実話だよ）。

なにしろ，「ミニ新幹線」である。これは，従来のいわゆるフル規格新幹線とは違って，一般列車用の軌道を広げて新幹線が走れるようにした一種の簡略版。コストもフル規格新幹線の10分の１ですむ経済的な列車だ。

しかも，東京－福島間は東北新幹線に引っ張っていってもらい，山形－福島間の自走区間は「スーパーひたち」などと同じ130キロしか出さない。まあ，名前だけが「新幹線」の妥協の産物だから，この程度のギミックはしかたがない。

そのせいか，最近巷では「だからミニ新幹線なんか……」という論調ができつつある。

しかし，これは非常にマズい。というのも，今後作られる予定のフル規格新幹線は，北陸新幹線を除き，すべて半永久的に採算割れするということが目に見えているからである。

たとえば，今後予定されている整備新幹線３線（北陸，東北延長分，九州）は，直接予算だけで１兆3500億円という工事費である。つまり，フル規格新幹線の建設費用は，距離にもよるが，１路線１兆円と見ていい。さらにランニングコストもひどい。数年前の段階で東北，上越とも，それぞれ年間1000億円の超赤字だったといわれている。年々乗客は増えているものの，事態が好転しているとは考えづらい。よしんば両線が黒字化しても，今後作る路線はなお地方を走る。

フル規格だとまず採算割れは確実で，年々貴重な税金や運賃をドブに捨てていくようなものなのだ。

ただし，一部には「やっぱり本物の新幹線がいい」という人もいるだろう。特に，新幹線が走る予定の地元の人にとっては，フル規格新幹線が走っている場所の人に対して負けたような気になるようだし，そういう地元民の意識を悪用する建設業者とか代議士の世論操作もある。

だから，地元の人も「もともと赤字路線の地方なんだから，ミニでもマイクロでも新幹線が走ってくれるだけで万々歳」というくらいの意識はもつべきだろうし，少なくとも都市部の一般市民は税金や運賃の用途まで考えた認識はもつべきだろう。

といっても，べつに僕は「地方には速い交通機関はいらない」とはカケラも思っていない。航空機をバンバン飛ばして，超豪華高速バスで補完すればいいだけのこと。それだけで問題は簡単に解決する。

もはや運賃格差もないし，「列車のほうがいい」というのは，単なる迷信にすぎない。「新幹線の駅が地元にできようものなら，もう一気にそこは国際文化都市」というのも妄想以外の何物でもない。

＊　＊　＊

と，ここまで書いてきて，今回も最後に申し訳程度にコンピュータの話が入る（すまない）。

交通機関以上にコンピュータは速いほうがいい。たまに「あまりに速すぎると問題がありそうだ」と勘違いしている人もいるが，ことコンピュータにかぎっては，絶対に「速は遅を兼ねる」はずなのである（注：たまに兼ねないこともある。だから某社の日本標準パソコンは，複数のCPUを搭載する）。

たとえば，ソフトを作ったりノウハウで性能をカバーできないような人の場合，あまりに性能が高いコンピュータだとまずいのではないか？　という議論がある。しかし，これは誤りだ。ソフトを自作できない人は，動作効率の悪いプログラムしか書けないし，運用ノウハウで劣る人がまっとうにそのマシンやソフトを使いこなすためには，もたもたした作業を吸収してくれるほどCPUが速くなくては，当然問題が出てくる。

同じことは，メモリ容量にもいえる。広いメモリ容量があれば，下手くそなプログラムでもゆうゆうとオンメモリで収容できる。

速いCPUに広いメモリ。

「夢の超特急」と同様に「夢のコンピュータ」の理想なのであろう。

ちなみに，新幹線が開通した頃のコンピュータは，８ビットパソコンよりも劣る機能だったそうだ。コンピュータが発展しすぎているのか，列車の進歩が止まっているのか。

料金受取人払

高輪局承認

1396

差出有効期間
平成6年7月
15日まで

郵便はがき

1 0 8 - 0 0

5 0 7

(受取人)

東京都港区高輪
2-19-13 NS高輪ビル

ソフトバンク株式会社

Oh!X 編集部行

□□□-□□ 電話

住所

氏名 年齢

職業・勤務先
学校・学部・学年

●編集部へのメッセージ

今月号の特集について

いちばん良かった記事

興味のなかった記事

これから載せてほしい記事内容

本誌以外にお読みのパソコン雑誌

推薦する市販ソフト

ソフト名：

推薦理由：

現在使っているコンピュータが遅いと感じるのは何をしているときですか？

あなたの愛機は（所有機種に○印をつけてください）　ない
X1（マニアタイプ,C,D,F,G,twin）X1 turbo（model 10,20,30,40,II,III,Z,ZII,ZIII）
MZ-（80K/C, 1200, 700, 1500, 80B, 2000, 2200, 2500, 2861）
X68000（初代,ACE,PRO,PROII,EXPERT,EXPERTII,SUPER,XVI,Compact, HD）
その他　MIDI楽器（　）
FD（　基）　TAPE　QD　HD（　MB）　MO　プリンタ（　）

年齢　歳　パソコン歴　年　男・女　プレゼントNo.

通常払込料金
加入者負担

払込通知票

口座番号	東京	1	□	29307	金額	
加入者名	ソフトバンク株式会社				料金	払込み / 特殊 円
払込人住所氏名	※ （郵便番号　　　）				備考	
					受付局日付印	

各票の※印欄は、払込人において記載してください。

記載事項を訂正した場合は、その箇所に訂正印を押してください。

この払込通知票は、機械で使用しますので、下部の欄を汚さないよう特に御注意ください。また、本票を折り曲げたりしないでください。（郵政省）

切り取らないで郵便局にお出しください。

通常払込料金
加入者負担

払込票

口座番号	東京	1	□	29307
加入者名	ソフトバンク株式会社			
金額				
払込人住所氏名	※			
備考		受付局日付印		

振替用紙

←点線から、きれいに切り取ってご使用ねがいます。

切り取り線

【定期購読のご案内】

●定期購読のお申し込みは、この郵便振替用紙のみとさせて頂き、銀行振込・現金書留によるご入金は、ご遠慮下さい。

●受付締切は、

1日発売	:	発売日前月10日振込
8日発売	:	〃　15日振込
15・18日発売	:	〃　25日振込

です。

〈例〉 4月1日発売（Oh！PC 4月15日号）の場合、お振込の締切は3月10日です。

締切に間に合わなかった場合は、自動的に次号からの発送となります。

なお、すでに発売されているもの、また、お振込が締切に間に合わなかった月号のものは、定期購読ではお求めになれません。書店でご購入ください。

●定期購読誌のお届けは書店発売日より遅くなりますのでご了承下さい。

「発売日一覧」

◇毎月1・15日発売
Oh！PC

◇毎月18日発売
Oh！X
Oh！FM TOWNS
C MAGAZINE
Oh！Dyna

◇毎月8日発売
月刊情報処理試験
LANTIMES
THE WINDOWS
DOS／Vmagazine
UNIX USER

切り取らないで郵便局にお出しください。

この欄は、加入者あての通信にお使いください。

送り先

お名前　フリガナ

お電話

ご住所　フリガナ
〒

定期購読申込書

□Oh！PC	年間	(23回)	12,880円	(新規／継続 NO.　　)
□Oh！PC	6ヶ月	(12回)	6,720円	(新規／継続 NO.　　)
□Oh！X	年間	(12回)	7,200円	(新規／継続 NO.　　)
□Oh！FM TOWNS	年間	(12回)	7,440円	(新規／継続 NO.　　)
□C MAGAZINE	年間	(12回)	11,760円	(新規／継続 NO.　　)
□Oh！Dyna	年間	(12回)	9,120円	(新規／継続 NO.　　)
□月刊情報処理試験	年間	(12回)	8,160円	(新規／継続 NO.　　)
□月刊情報処理試験	6ヶ月	(6回)	4,080円	(新規／継続 NO.　　)
□LANTIMES	年間	(12回)	17,760円	(新規／継続 NO.　　)
□THE WINDOWS	年間	(12回)	11,760円	(新規／継続 NO.　　)
□DOS／Vmagazine	年間	(12回)	9,360円	(新規／継続 NO.　　)
□UNIX USER	年間	(12回)	11,760円	(新規／継続 NO.　　)

〔備考欄〕

この払込通知票は、機械で使用しますので、下部の欄を汚さないよう特に御注意ください。また、本票を折り曲げたりしないでください。　（郵　政　省）

愛読者プレゼント

コナミ　☎03(3264)5678

1 グラディウスII

X68000　5″2HD版2枚組

9,800円(税別)　**3名**

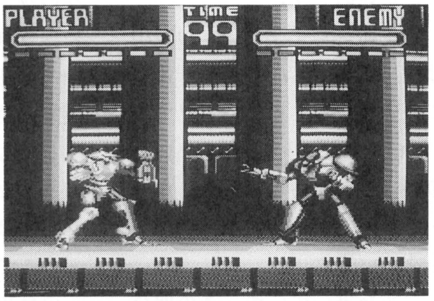
X68000といえば「グラディウス」，というのはもはや昔のことで，いまはX68000といえば「グラディウスII」だ！　この超人気伝統的横スクロールシューティングゲームを3名の方に。

ブラザー工業　☎052(824)2493

2 ヘビーノヴァ

X68000用　3.5/5″2HD版

5,800円(税別)　**5名**



流行の格闘ゲームでは，筋肉モリモリのお兄ちゃん（お姉ちゃんもいるけど）が出てくるものが多いみたい。しかし，この「ヘビーノヴァ」はロボットどうしで争う，対戦型格闘ゲームなのである。

セミック　☎03(3582)6821

3 電子手帳用 フラッピーカード

8行表示専用　6,000円(税込)

2名

「フラッピー」，昔からのパソコンゲームファンには忘れられないアクションパズルゲームだ。その「フラッピー」が電子手帳用のカードになった。ただし，8行表示のシャープ電子システム手帳専用なのでご注意を。

プレゼントの応募方法

とじ込みのアンケートはがきの該当項目をすべてご記入のうえ，希望するプレゼント番号をはがき右下のスペースにひとつ記入してお申し込みください。締め切りは1992年9月18日の到着分までとします。当選者の発表は1992年11月号で行います。

4 テレホンカード各種

4名

各方面からいただいたテレホンカードを1枚ずつ各1名に差し上げます。どれが誰に送られるかはこちらで適当に選ばせていただきますので，ご了承のほどを。

5 上昇気流 vol.3

10名

またまた「上昇気流」が余ってしまったので放出します。……なんてことを書くと発行人の高橋哲史くんに怒られてしまうかな。では，もう一度最初から。本誌スタッフである高橋哲史氏のご厚意により，ミニコミ誌「上昇気流vol.3」を特別に10名の方にお分けします。

7月号プレゼント当選者

1 レミングス　(埼玉県)川井健司　鈴木宗太郎　(大阪府)石垣直樹　**2** ノア　(青森県)古川圭一　(千葉県)濱田研一　(神奈川県)花上康典　**3** XIN/XOUT II　(福岡県)和田岳雄　**4** ウィザードリィ幻想曲　(愛知県)近藤匡紀　(奈良県)中西厚　**5** Inside X68000　(神奈川県)八木明　(香川県)黒川孝造　(敬称略)

以上の方々が当選しました。おめでとうございます。商品は順次発送いたしますが，入荷状況などにより遅れる場合もあります。また，雑誌公正競争規約の定めにより，当選された方はこの号の他の懸賞には当選できない場合がありますのでご了承ください。

近未来型ワープロ序説

明日〜2年後

僕は，ふだんワープロはMacintosh用の「MacWriteII」というのを使っています。特にこれが格段にいいという理由があるわけではないのですが，ほかのいくつかのワープロに比べて，このソフトを作った人のセンスのよさが全般に感じられるので，一応信頼して使っているのです。

使っている時間が長いということもあって，細々したことも含めて，いろいろともっとこうあってほしいという望みがどんどん出てきます。まずここで，そのような要望願望の山の中から3つほど，どちらかといえば細かい話に属することについて書いてみましょう。もし，もうどこかのワープロで実現されているようでしたら教えてください。

メモ書きシート

紙に手書きした場合と比べて，ワープロ印字はどうしてもお行儀がよくなってしまうので，書き手の生の感情というものが伝わりにくくなります。たとえば，大事なところをまるく囲ったり，赤線を引いたり，あるいは補足を書き込んだり，少し字を大きく書いたりするような自然な表現が，ワープロでは限定を受けてしまいます。もちろん，機能的にはワープロでも実現可能ではありますが，やはり，いまいち書き手の感情が伝わりにくいという面は否定できないでしょう。

そもそも，手書きだと重点を置きたい部分には自然と力がこもるので，何もしなくてもそれなりの情報（ここが重要なのだ）が読み手に伝わるというものです。こういう，どちらかといえばお行儀の悪いといえるような部分こそが重要な付加的情報なのだと，僕はいいたいのです。

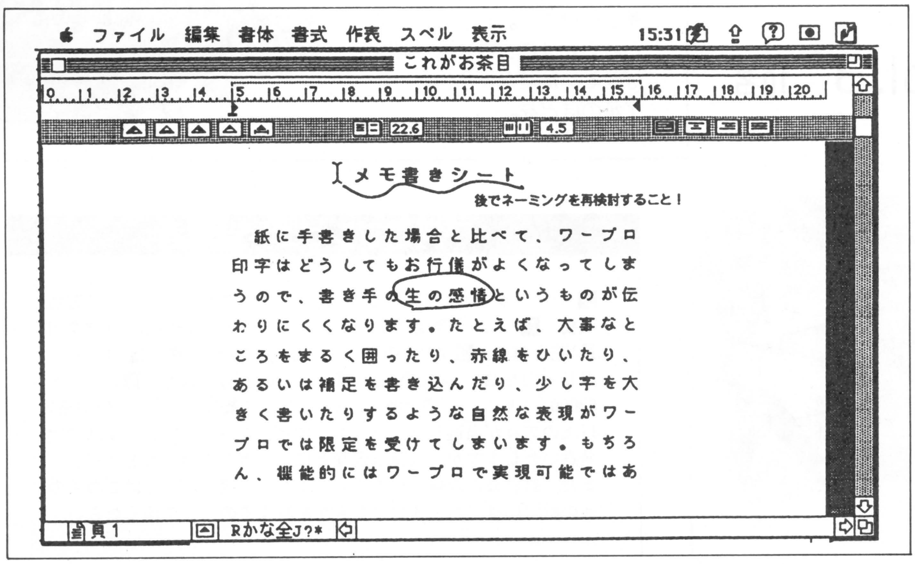
ここでは，いま述べたような印字の表現力については横に置いておいて，少し違った角度からこのようなことについて見てみましょう。「メモ書きシート」とでもいいましょうか，印刷はされないけれども，自分にとってあとで参考となる情報を付け加える機能のことをいいます。

たとえば，文章を書いていて，この部分はあとでもう1回推敲が必要だと思ったら，その部分を赤いペンでまるく囲っておき，「要推敲」とか書いておきます。あるいは，自分が気に入った表現があったなら，青色で波線をラフにそのところに引いておき，「こりゃ，いい表現！」などと書きます。

このように，出来上がりの文章をよくするためでなく，自分のために付加的な情報をお絵描きのように自由なフォーマットで残しておく機能です。

イメージとしては，文書の紙の上に透明なシートがぴったりくっついており，その上に色付きのサインペンで自由に絵や字を書けるといった感じ，あるいは，文書の上に自由に付箋（ポストイット）が貼れるといった感じでしょう。

図1　メモ書きシート例



タイミング情報利用型FEP

“かな漢字変換をどのようにすればベストなのか？”という問題は，日本語ワープロに課された大きな宿題です。これまでいろいろな変換法が開発されてきました。が，漢字の選択にはユーザーの主観や嗜好がどうしても入る，つまり100%正解というのはありえない問題ですので，これが絶対という方法は生まれないのでしょう。

ここでは，変換キーを押すというユーザーの負担を減らすことを考えたときに，重要と考えられるひとつの技法について述べたいと思います。例として，「お前歯1本が痛い」という少し変な文章を考えてみます。もし，「おまえ＋変換キー＋はいっぽんが＋変換キー＋いたい」と入力したならば，たぶんどんなフロントエンドプロセッサ(FEP)でも，思いどおり漢字が出るまでにそう余計なキー入力を必要としないでしょう。

しかし，いちいち変換キーというのも余分といえば余分な負担です。「おまえはいっぽんがいたい」と入力しても，きちんと変換してほしいという要望が当然あります。しかし，多くのFEPの場合，「お前は1本が痛い」となってしまうでしょう。こちらのほうが構文的にはふつうですから。

ここで，ユーザーがキーを押す間隔タイミングに関する情報を使うのです。要するに，ユーザーが文章を入力するとき，キーを打つタイミングは構文，あるいは意味の切れ目に対応しているのではと期待するわけです。

この例では，キーを打つ間隔が長くなる部分が，「おまえは＾いっぽん」ではなく「お

まえ＾はいっぽん」のほうにあるのではないかと予想されるのです。この情報を使うことによって，このような漢字変換もうまくいくようにさせようという話です。ここで重要なのは，ユーザーに意識させてキーを押す間隔を調整させるのではなく，ごく自然に出現するタイミング情報を汲み上げて，漢字変換の際のデータのひとつに使おうという点です。

もちろん，このタイミング情報は万能ではありません。いつも意味の切れ目にマッチするタイミングになるとは限りませんが，変換の候補順を決める有力な情報になりうるのではないかということです。

もし，「おまえはいっぽんがいたい」という字づらだけから，正解の漢字を選び出そうとするには，「１本」や「痛い」に関するなんらかの意味的情報が必要になるでしょうから，それをまともにやるのは，たいへんすぎるのではないかと思われるのです。

マクロ操作自動設定

ワープロを操作しているとき，基本的なキー操作の繰り返しというものは意外と出てくるものです。しかも，わざわざあらたまって定義するほどいつも出てくるというものでもないような性格のものの繰り返しです。たとえば，印字出力を整えるために，タブを行頭に挿入し，カーソルを右に３語分移動して，……といった具合です。

このような一連のキー入力を２，３回繰り返した場合，規則性を機械が見つけて，どこまでこの処理を繰り返すのかを確認したあと，自動的にやってほしいのです。

本当は，もっと複雑な規則，たとえば数字で始まる行はある処理をし，文字で始まる行は別の処理をして，のように，条件に応じた判断も自分で認識して，その先を実行するようになってほしいのです。

この機能を実現するには，かなり高度な知的推論機構が必要となると思われます。現在すぐにできることは，最近行った操作の履歴がメニューとして出てきて，好きな部分をワンタッチでマクロ化することくらい。これで，まあ，よしとしましょうか。

2年後〜10年後

ここまでに述べた３つの話より，もっとワープロの基本的なポリシーに関わる話を以下に４つだけ述べてみます。本人は真面目なのですが，半分冗談に思える話もあるかもしれません。

アクティブデータベース型ワープロ

ワープロが動いているときに，FEPだけでなく各種のデータベースがフル稼働していて，文章作成を支援してくれるような総合的な環境が実現すると，それには思わぬご利益があるのではないでしょうか？　もはや，ワープロは単なる文字列作成のツールから大きく飛躍して思考ツールとなるのです。

ワープロで文章を入力していると，「それ!　入力」として，各種のデータベースが探索され，各ウィンドウにその結果がリアルタイムに表示されるということなのです。辞書，類義語辞典だけでなく，表現データベース，人名データベースなど多種多様です。

重要なのは，データベースの探索だけでなく，登録も半自動的に行われるということです。ですから，少し気に入った表現があれば，そのフレーズがどの文書で使われているという情報付きで，ワンタッチで格納されるというわけです。

実は，この環境は我々の頭の中でいつも起こっていることそのものともいえます。ある言葉を意識した瞬間に連想が広がります。これは，データベースの並列探索にほかならないわけです。これを計算機の画面上で行えば，というのです。

僕自身，このような機能が実現された環境がどこまでパワフルなのか，まだはっきりとしたイメージをもってはいません。しかし，いままでのワープロを遙かに超越するような，想像を絶する知的道具になるのではないかと期待します。

このワープロで大事なのは，パーソナルなデータベースとの有機的結合です。単に既成の辞書類が絶えず探索されているというのではなく，自分自身にとって意味のある情報をほかの情報とリンクをはかりながら記憶させて随時呼び出すというところが決め手なのです。

グループワープロ

アップル→ネクストのスティーブ・ジョブスのいう「インターパーソナルコンピューティング」という概念を実現するひとつの有力な道具として，僕は「グループワープロ」というものを考えています。これは，複数の人のあいだであるテーマについてネットワーク上で議論し，その結果に基づいてひとつの文書をまとめるまでをサポートするというものです。

ネットワーク上で議論するためのシステムとして，電子掲示板，あるいはニュースシステムというものがすでに存在します。これは，単に相手の記事を引用しながら，自分の記事を投稿し，それを皆で読むというものです。このシステムをベースとし，さらに合意がとれた部分を少しずつ文書として作り上げていくという機能をもたせて拡張したものである，といったら理解していただけるでしょうか。

ローカルなニュースグループを僕自身も使っているのですが，どうも記事が投稿されっぱなしで，ただ溜まっていくといった傾向が強いのです。そうではなく，せっかくの議論ですから，なんらかの形でまとまった文書になってほしいと思うのです。そこで，そのような方針で新たにグループワープロという概念をもつシステムを考えようというのです。

このシステムでは，作り上げていく文書の構造をある程度作るという作業がまず必要になります。その後は個別な論点を設定し，話し合い，結論をまとめてはその結果を文書構造の適当な位置に挿入するという操作をすることになります。

文書は，表面的には議論の結果の整理さ

近未来型ワープロ序説

れたまとめとして見えますが，該当する議論の部分にもリンクが貼られており，あとからなぜこういう結果に落ち着いたのかを追うこともできるようになっています。

自然言語を許さないワープロ

ワープロで入力した文書というものには，きれいにプリントアウトして他人に伝えるための情報という側面もあれば，自分自身の表現したものを記録した情報という側面もあります。現在のところ，ワープロの一般的な利用のされ方を見てみると，前者のほうの色合いのほうが濃いように思われます。しかし，これからワープロがもっと深い知的情報処理を行うようになった場合に重要となるのは，後者のほうでしょう。

つまり，きれいにプリントアウトすることよりも，データとしてとっておいて，それをいろいろな方法であとあとに有効利用するほうが重要になってくるのではないかということです。

そこで問題になるのが，見せる文書としての側面と処理されるデータとしての側面とのギャップです。文書として書かれた自然言語を，機械がその意味を完全に把握できるのならば問題はないといえましょう。自動的に処理可能なデータに変換すればよいからです。

しかし，それを完全に行うのは本質的に無理な注文かもしれません。もともと，あいまいな文章だって人間は書いてしまうのですから。

そこで，あえてここでは，僕は従来とは逆のアプローチをとります（案外短気なので）。それは，機械が人間の言葉を理解する日を待つのではなく，我々がデータ形式に合った方法でワープロに情報を入力するのです。そして，文書化は機械に自動的にやらせようというアプローチです。

このように，人が表現したいことを一定のルールに従って入力し，機械が用途に合わせて書式や文体などを考慮して出力するという方法の利点は次のようなことです。

1）日本語には特に起こりやすいあいまいな表現がなくなる。また，文章の意味的な間違いもリアルタイムにチェックできる。

2）機械が意味を完全に把握するので自由に処理ができる。たとえば，リアルタイムに各国語の文章が作成できる。

3）書かれた情報のデータベース化が容易で，有効利用できる。

ステータスワープロ

ワープロを使うことによって，それ以前よりもっといい文章を書けるようになり，また(こっちのほうが重要なのですが)，もっといいアイデアが生まれるようになることが理想なのですが，実際にはそのようになっていない場合が多いような気がします。

特に（ワープロに不慣れなために，あるいは機能不足で），文章の編集がままならない状態の人においてよく見られるのですが，どうも文章が荒っぽいまま残りがちな気がします（おっとー，人のことはいえんが）。

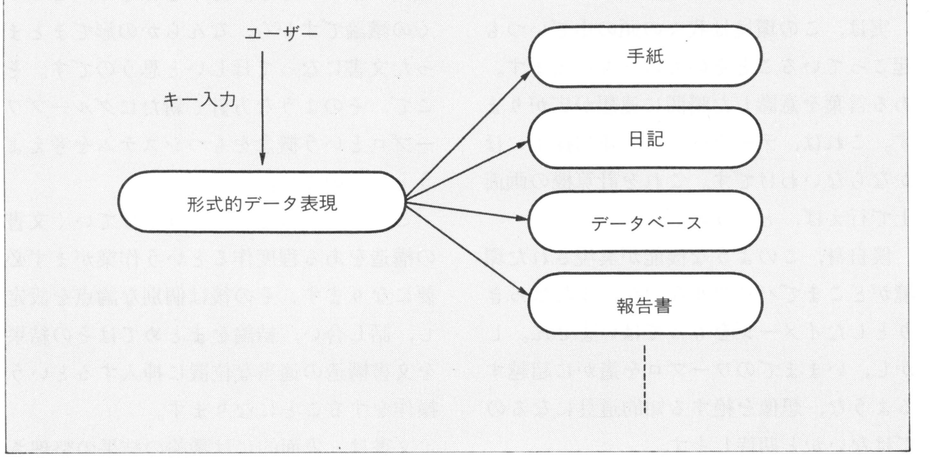
知的情報処理が実現された将来において，ワープロに望むのは，よりよい文章を作る能力を養成する機能です。ひと言でいえば，悪文が書けないようなワープロというものを望みたいのです。

ワープロの機能がもっと進んでくると，ワープロにもユーザーを選択する権利が発生するかもしれません（なんじゃこりゃ，ちょっと苦しいかな？)。あまりにも目に余る文を入力しすぎたり，せっかく用意された機能を使わないでいると，しまいに「あなたはこのワープロを使用するレベルに達していません。購入された店で払い戻しを受けてください」と画面に出て，もう動かなくなるというものです。

このような機能は何を意味するのでしょうか？　要するに，これは新たなステータスの誕生を意味するのです。「僕はXワープロを使用しています」と公衆の面前で述べると，「ウォー！」というどよめきがわき起こるのです。

知能情報化社会においては，知的道具を駆使して素晴らしい文章を作る能力が，従来の古めかしいライセンスやステータスを遥かに超える評価を受けるようになるのです。

図2　自然言語入力を許さないワープロ



筆者より

今回はいろいろ考えている近未来ワープロ像の断片のみを書きました。そのうちまとまった形になればいいと思っています。皆さんも，ありきたりでなく，過激で革新的なワープロを無責任でもいいから思いついたら，アンケートハガキにでも書いて編集部まで送ってください。紹介します。また，万一興味をもたれた研究者の方などがいらっしゃったのなら，junet: ari@debris.elcom.nitech.ac.jpまでご連絡ください。

Oh!X LIVE in '92

X68000・Z-MUSIC用 恋をしようよ Yeah! Yeah!

Minami Masaki
南　正樹

X68000・Z-MUSIC用 ゆめいっぱい

Tadokoro Hiroyuki 田所 広行

今月は，最近LIVE inでは常連となりつつあるLINDBERGの曲と，あのちびまる子ちゃんのオープニングソングをお届けします。2曲ともX68000・Z-MUSIC用です。さあ，張り切って入力してくださいね。

恋をしようよ！

さて，今月の１曲目はLINDBERGのアルバム「LINDBERG V」のなかから，「恋をしようよ Yeah! Yeah!」をお届けしましょう。X68000用で，Z-MUSICシステムが必要です。この曲はコカ・コーラのCMソングにもなっていますので，サビくらいは聞いたことがあるでしょう。LINDBERGは，このLIVE inにも過去に２回登場していて人気のほどがうかがえます。投稿してくれた南君もLINDBERGのファンらしく，歌詞までていねいに書いて送ってくれました。

曲はとってもグーです。最初のギターといい，ノリといい，なかなかよくまとまっています。プログラムが長いのが難点といえば難点ですが，過去にはもっと長いものも載っていますし，問題ないでしょう。

ただ，PCMファイルが120Kバイト程度になりますので，Z-MUSICの設定によっては演奏できないかもしれません。－Pオプションで150Kバイト程度のPCMバッファを確保しておけば大丈夫でしょう。

ワークエリアをいっぱい取るのもばかばかしいので，ZPCNV.Xを使って.ZPDファイルを先に作っておいてください。プログラムでもそういった設定になっています。

PCM8.Xを使っていないので，PCM8を持っていない人でもOKですが，そのぶんPCMファイルが大きくなったといったところでしょうか。

南君はパソコン通信でもやっているのでしょうか。非常に見やすいドキュメントファイルでありがたかったです。できれば，作ったときの苦労話やセールスポイント，自己紹介なんかをいっぱい書いておいてくれると，もっとよかったんですけどね。

笑顔の魔法をおしえて

日曜の夕方はテレビの前に座っている人も多いのではないでしょうか。もちろん，見ているものは「ちびまる子ちゃん」ですよね(読者の年齢層を無視した発言)。ちょっと懐かしくて，なんとなく笑ってしまえる(身に覚えがある？)，ほのぼのとしたアニメです。このちびまる子ちゃんの曲といえば，「おどるポンポコリン」が超有名ですよね。っていっても，今回お届けする曲はオープニングテーマの「ゆめいっぱい」のほうです。

演奏にはZ-MUSICシステムのほかに，SC-55が必要です。SC-55からしか音は出ませんので，ミキシングの必要はありません。

テレビサイズということで，１コーラスしかありませんが，余計にテレビを見ているような錯覚に陥ることができます。リストも短いので，入力しやすいのではないでしょうか。デキのほうはちゃんと保証します。あの楽しい雰囲気を見事に再現していますよ。

プログラム中で，パート３や14などでループの中にバンク切り替えコマンドがありますが，これは１回だけだとバンク切り替えされないことがあるためだそうです。SC-55に詳しい人で，この症状の原因がわかる人がいたら教えてください。

この作品は音取りでなかなか苦労したそうです。とにかく，楽譜が高かったそうで，「コードをお店でメモってしまう」という必殺ワザを使ったとか。ひとりがコードを読み，ひとりがメモり，ひとりがそれを隠すというコンビネーションも完璧です。まあ，次回からは録音できるウォークマンかテレコを使うことをお勧めします。

初投稿ということですが，これからも常連目指してどしどし投稿してくださいね。

(S.K.)

LINDBERG

ちびまる子ちゃん

リスト１　恋をしようよ Yeah! Yeah!

日本音楽著作権協会(出)許諾第9270994-201号

```
.comment        恋をしようよ Yeah! Yeah!  < LINDBERG >
.comment        作詞 渡瀬マキ/作曲 川添智久

.comment        PROGRAM  南 正樹/1992-05-25

/ for ZMUSIC.X

/-----------------------------------------------------
/ TRACK SETUP

(i)

(m1,1200)(aFm5,1)    /Vo1
(m2,1200)(aFm6,2)    /Vo2
```

```
(m3,1200)(aFm7,3)    /Ch,Vo
(m4,2100)(aFm4,4)    /Gt1
(m5,3600)(aFm1,5)    /Gt2
(m6,1200)(aFm7,6)    /Gt3
(m7,1000)(aFm5,7)    /Kb
(m8,2400)(aFm8,8)    /Ba
(m9,2000)(aAdpcm,9) /Dr
/--------------------------------------------------------------
/ ADPCM DATA SET
.ADPCM_BLOCK_DATA=Yeah.ZPD
/--------------------------------------------------------------
/ OPM DATA SET
(v19,0  /        F Guitar
/          AF  OM  WF  SY  SP PMD AMD PMS AMS PAN
           57, 15,  2,  0,200,  0,  0,  0,  0,  3,  0
/          AR  DR  SR  RR  SL  OL  KS  ML DT1 DT2 AME
           31, 22,  8,  6,  7, 11,  2, 12,  6,  0,  0
           31,  6,  0,  6,  3, 33,  1,  3,  3,  0,  0
           28,  6,  0,  6, 15, 32,  0,  3,  4,  0,  0
           31,  8,  0,  8, 15,  0,  0,  1,  4,  0,  0)
(v24,0  /        E Guitar 4
/          AF  OM  WF  SY  SP PMD AMD PMS AMS PAN
            2, 15,  2,  0,200,  0,  0,  0,  0,  3,  0
/          AR  DR  SR  RR  SL  OL  KS  ML DT1 DT2 AME
           28,  0,  0, 10,  0, 57,  0,  2,  7,  0,  0
           31, 18,  0, 10,  2, 33,  1,  8,  7,  0,  0
           26, 16,  6, 10,  2, 29,  1,  0,  4,  0,  0
           28,  6,  0,  8, 15,  0,  1,  1,  4,  0,  0)
(v28,0  /        E Bass 1
/          AF  OM  WF  SY  SP PMD AMD PMS AMS PAN
            3, 15,  2,  0,200,  0,  0,  0,  0,  3,  0
/          AR  DR  SR  RR  SL  OL  KS  ML DT1 DT2 AME
           31, 12,  0, 10, 15, 47,  0,  5,  6,  0,  0
           31,  0,  0, 10,  0, 23,  0,  0,  4,  0,  0
           31,  0,  4,  6,  0, 33,  0,  0,  4,  0,  0
           28,  0,  6,  8,  0,  0,  0,  0,  3,  0,  1)
(v51,0  /        E Organ 6
/          AF  OM  WF  SY  SP PMD AMD PMS AMS PAN
           62, 15,  2,  1,190, 10,  0,  1,  1,  3,  0
/          AR  DR  SR  RR  SL  OL  KS  ML DT1 DT2 AME
           31,  0,  0, 15,  0, 30,  0,  0,  3,  0,  0
           31,  0,  0, 15,  0,  0,  0,  0,  7,  0,  1
           31,  0,  0, 15,  0,  0,  0,  3,  2,  0,  1
           31,  0,  0, 15,  0,  0,  0,  2,  3,  0,  1)
(@1,    /        Vocal
/          AR 1DR 2DR  RR 1DL  TL  RS MUL DT1 DT2 AME
           31,  0,  0,  0,  0, 39,  1,  6,  3,  0,  0
           31,  3,  1,  1,  1, 38,  1,  7,  3,  0,  1
           19,  2,  1,  6,  1, 38,  1,  1,  7,  0,  0
           16,  0,  0,  9,  0,  0,  1,  2,  7,  0,  1
/          AL  FB  SM PAN
            0,  8)
(@2,    /        Synth
/          AR 1DR 2DR  RR 1DL  TL  RS MUL DT1 DT2 AME
           25, 18,  0, 15,  0, 42,  0,  1,  5,  0,  1
           28,  4,  0, 15,  0,  2,  0,  2,  3,  0,  1
           20, 15,  0, 15,  0, 35,  0,  0,  3,  0,  1
           28, 15,  0, 15,  0,  2,  0,  1,  3,  0,  1
/          AL  FB  SM PAN
            7,  7)
(@82,   /        D.G H (Oh!X 1991/12)
/          AR 1DR 2DR  RR 1DL  TL  RS MUL DT1 DT2 AME
           27, 25,  1,  9,  2, 27,  0,  4,  3,  2,  0
           27, 20,  1,  9,  4, 24,  1, 14,  6,  0,  0
           31,  2,  1,  9,  3,  7,  0,  0,  3,  0,  0
           30, 14,  0,  9,  0,  1,  1,  1,  7,  0,  0
/          AL  FB  SM PAN
            3,  7)
/--------------------------------------------------------------
/ MML DATA SET
/     Vo1 (Fm5)
/[A]
(t1) @k+4 v11 l8 q7 @1
(t1) |:10r1:| r1
/[B]
(t1) @1 |: o5 cc4c4>gg4 a4aag4&ff16.&(f32g)&g&q8e4q7rr2
(t1) |1 r1 :| |2 @k0p1v12 |:3r1:| <d1 r1 @k+4p3v11:|
/[C]
(t1) @1 |: o4c2c4ed& d4r2. f4e4d4c4 >b4<c4d4>b4
(t1) <ccccc4ed& d4r2. f4e4d4c4 |1 >b4b<c4.d4 :|
(t1) |2 a4g4e4d4 c1 r1 :|
/[D]
(t1) |:  v11 l8 q7 @1 o4 r2aaaa ag4rr2 r2aaaa
(t1) a4.gg2 r2araa ag4rr2 a2.aa& b2.r4
/[E]
(t1) |:o5cc4c4>gg4 a4aag4&ff16.&(f32g)& g&q8e4q7rr2 r1:|
(t1) f4.<c4.r4 >eee<c4.r4 >f4f<c4.c4 >b4bg4.r4
(t1) a2&agag g2r2 a2.a4 b4g4b<d4c&
/[E1]
(t1) |1  c2.r4 |:7r1:|
/[C']
(t1) @1 o4 c4c4c4ed& d4r2. f4e4d4c4 >b4<c4d4>b4
(t1) <ccccc4ed& d4r2. f4e4d4c4 a4g4e4d4 c1 r1  :|
/[E2]
(t1) |2  o5 c2.r4 |:3r1:|
/[F]
(t1) |:19r1:|  |:3r1:| o5 @k0p1v12@1 d1 r1 @k+4p3v11 r2
/[G]
(t1) v11 l8 q7 @1
(t1) o5 cc4c4>gg4 a4aag4&ff16.&(f32g)& g&e4rr2 r1
/copy と書いてある所はコピーできます
/[E]copy
(t1) |:o5cc4c4>gg4 a4aag4&ff16.&(f32g)& g&q8e4q7rr2 r1:|
(t1) f4.<c4.r4 >eee<c4.r4 >f4f<c4.c4 >b4bg4.r4
(t1) a2&agag g2r2 a2.a4 b4g4b<d4c&
/[E3]
(t1) c2.r4 r1
/[H]
(t1) |:o5cc4c4>gg4 a4aag4&ff16.&(f32g)& g&q8e4q7rr2 r1:|
/[I]
(t1) |:4o5cc4c4>ggv9@k0<g gg4a4.&g4@k+4v11 |:r1:|:|r1 :|
/--------------------------------------------------------------
/     Vo2 (Fm6)
/[A][B][F]以外は(t1)と同じです
/[A]
(t2) @k-2 v11 l8 q7 @1
(t2) |:10r1:| r1
/[B]
(t2) @1 |: o5 cc4c4>gg4 a4aag4&ff16.&(f32g)&g&q8e4q7rr2
(t2) |1 r1 :| |2 @k0|:r1:| b2.b4 b1 r1 @k-2:|
/[C]
(t2) @1 |: o4c2c4ed& d4r2. f4e4d4c4 >b4<c4d4>b4
(t2) <ccccc4ed& d4r2. f4e4d4c4 |1 >b4b<c4.d4 :|
(t2) |2 a4g4e4d4 c1 r1 :|
/[D]
(t2) |:  v11 l8 q7 @1 o4 r2aaaa ag4rr2 r2aaaa
(t2) a4.gg2 r2araa ag4rr2 a2.aa& b2.r4
/[E]
(t2) |:o5cc4c4>gg4 a4aag4&ff16.&(f32g)& g&q8e4q7rr2 r1:|
(t2) f4.<c4.r4 >eee<c4.r4 >f4f<c4.c4 >b4bg4.r4
(t2) a2&agag g2r2 a2.a4 b4g4b<d4c&
/[E1]
(t2) |1  c2.r4 |:7r1:|
/[C']
(t2) @1 o4 c4c4c4ed& d4r2. f4e4d4c4 >b4<c4d4>b4
(t2) <ccccc4ed& d4r2. f4e4d4c4 a4g4e4d4 c1 r1  :|
/[E2]
(t2) |2  o5 c2.r4 |:3r1:|
/[F]
(t2) |:19r1:| @k0|:r1:| o4 b2.b4 b1 r1 @k-2 r2
/[G]
(t2) v11 l8 q7 @1
(t2) o5 cc4c4>gg4 a4aag4&ff16.&(f32g)& g&e4rr2 r1
/[E]copy
(t2) |:o5cc4c4>gg4 a4aag4&ff16.&(f32g)& g&q8e4q7rr2 r1:|
(t2) f4.<c4.r4 >eee<c4.r4 >f4f<c4.c4 >b4bg4.r4
(t2) a2&agag g2r2 a2.a4 b4g4b<d4c&
/[E3]
(t2) c2.r4 r1
/[H]
(t2) |:o5cc4c4>gg4 a4aag4&ff16.&(f32g)& g&q8e4q7rr2 r1:|
/[I]
(t2) |:4o5cc4c4>ggv9@k0<c dd4e4.&d4@k-2v11 |:r1:|:|r1 :|
/--------------------------------------------------------------
/     Ch,Vo3 (Fm7)
/[A]
(t3) |:10r1:| r1
/[B]
(t3) v10 l8 q7 @1
(t3) @1 |: o5 ee4e4cc4 >a4aag4&ff16.&(f32g)& g&q8e4q7rr2
(t3) |1 r1 :| |2 r2..p2v11a& g1& g2.g4 g1 r1p3v10 :|
/[C]
(t3) @1 |: o4c2c4ed& d4r2. f4e4d4c4 >b4<c4d4>b4
(t3) <ccccc4ed& d4r2. f4e4d4c4 |1 >b4b<c4.d4 :|
(t3) |2 a4g4e4d4 c1 r1 :|
/[D]
(t3) |:  v10 l8 q7 @1 o5 r2cccc c>b4rr2 r2aaaa
(t3) a4.gg2 <r2crcc c>b4rr2 a2.aa& b2.r4
/[E]
(t3) |:o5ee4e4cc4 >a4aag4&ff16.&(f32g)& g&q8e4q7rr2 r1:|
(t3) f4.<c4.r4 >eee<c4.r4 >f4f<c4.c4 >b4bg4.r4
(t3) <c2&c>b<c>b b2r2 a2.a4 b4g4b<d4c&
/[E1]
(t3) |1  c2.r4 |:7r1:|
/[C']
(t3) @1 o4 c4c4c4ed& d4r2. f4e4d4c4 >b4<c4d4>b4
(t3) <ccccc4ed& d4r2. f4e4d4c4 a4g4e4d4 c1 r1  :|
/[E2]
(t3) |2  o5 c2.r4 |:3r1:|
/[F]
(t3) |:19r1:| @1 o4 r1p2v11 g1& g2.g4 g1 r1 r2p3v10
/[G]
(t3) v10 l8 q7 @1
(t3) o5 ee4e4cc4 >a4aag4&ff16.&(f32g)& g&e4rr2 r1
/[E]copy
(t3) |:o5ee4e4cc4 >a4aag4&ff16.&(f32g)& g&q8e4q7rr2 r1:|
(t3) f4.<c4.r4 >eee<c4.r4 >f4f<c4.c4 >b4bg4.r4
(t3) <c2&c>b<c>b b2r2 a2.a4 b4g4b<d4c&
/[E3]
(t3) c2.r4 r1
/[H]
(t3) |:o5ee4e4cc4 >a4aag4&ff16.&(f32g)& g&q8e4q7rr2 r1:|
/[I]
(t3) |:4 o5 ee4e4ccv9>g aa4b4.&a4v10 |:r1:| :| r1  :|
/--------------------------------------------------------------
/     Gt1 (Fm4)
(t4) t187 v11 l8 q8
/[A]
(t4) @82 o5 c2.>g4 f4.e4.d&(d,o5c)& c2.>g4 f4.g&g2
```

▶大学院試まであと１カ月半。それにしても，受験がこんなにしんどいとは……。高校生のときは，結構がんばれたのになあ。　　松田　英弘(21)京都府

```
(t4) c1 >(b,o4e)&e2.. (cg)&g4(cb-)&(b-4a)&a4
(t4) (a16f)&f2... (g4a)&(a4b-)&(b-4,o5d-)&d-4
(t4) f1 >(a,o5e)&e2..
/[B]
(t4) |: v11q4 @19o3g<ec>g<gc>gd f<fc>f<gd>g<g
(t4) rec>g<ec>gd |1 r1:| |2 |:4v9@82r1:|
(t4) o4'd#4>g#4''e4>a4''e#4>a#4''f#4>b4'
/[C]
(t4) @51 v14 q8 o3 |:|:5r1:| r2
(t4) p1v3(ga)&v5(ab)&v6(b,o4c#)&v9(c#d#)&
(t4) (d#f)&p2(fg)&g4&g2& v8g2&v7g2:| r1 r1
/[D]
(t4) |:  |:8r1:|
/[E]
(t4) |: v11@19q4 o3g<ec>g<gc>gd f<fc>f<gd>g<g
(t4) rec>g<ec>gd r1 :| @82 o3  v9 ffffr2 eeeer2 ddddr2
(t4) <gr4g&g2>ffffr2 v7er4e4.v10<(fg)&g ggf4gf4. g4gg&g2
/[E1]
(t4) |1  c1& c2.r4 r1 v11 o5 p1 q8 c2.>g4
(t4) f4.e4.d4 r1 <c2.>g4 f4.g&g2 p3
/[C']
(t4) |:@51 v14 q8 o3
(t4) r1 r2p1v3(ga)&v5(ab)&v6(b,o4c#)&v9(c#d#)&
(t4) (d#f)&p2(fg)&g4&g2& v8g2&v7g2:| r1 r1  :|
/[E2]
(t4) |2  |:3r1:| o6 v11 r2r4.c&
/[F]
(t4) q8l16cde-dc>b-ga-b-gagfgl8(f16g)&g16q7

(t4) g2&(gf)&ff.f16&
(t4) f<(e-,o7e-)&e-4>(e-,o7e-)&e-4&(e-,o6e-)>
(t4) (e-2.,o4e-)<r(a16,o6f)&(f16g)&gb-f|:3(f16g)&g16:|ff
(t4) &(fg)&ga-(fg)&gg(e-f)&f e-(e-d)&d(e-16d)&d16cdc>b-
(t4) (g16f)&f16e-(c4,o4c)r>b-<ce-
(t4) (c16g)&g16b-<cd4.>>q2g4q7
(t4) l16|:o2(b-,o3c)&c:|(e-f)&fe-8e-(fg)&gfggb-g
(t4) (b-,o4c)&cc>b-<ce-ce-|:(e-f)&f:|g8(b-g)&g
(t4) l8b-<(e-16f)&f.(e-16f)&f16&(f4,o4c)>b-b-
(t4) <e-ce-f(f16g)&(g16f)&(f16e-)&e-16(fg)&g
(t4) e-(fg)&gf(f16g)&g16b-r>g <c>b-gb-4<c>b-4
(t4) gd4f(g2d)& (d1,o2g) r1 <(f1a)& a1

(t4) p3 |:24g:| (edcdc>b)1 r2
/[G]
(t4) v11@19q4o3g<ec>g<gc>gd f<fc>f<gd>g<g rec>g<ec>gd r1
/[E]copy
(t4) |: v11@19q4o3g<ec>g<gc>gd f<fc>f<gd>g<g rec>g<ec>gd
(t4) r1 :| @82o3v9 ffffr2 eeeer2 ddddr2 <gr4g&g2 >ffffr2
(t4) v7er4e4.v10<(fg)&g ggf4gf4. g4gg&g2
/[E3]
(t4) |:2r1:|
/[H,I]
(t4) v11 q8 |:6o5 c2.>g4 f4.e4.d4
(t4) <c2.>g4 f4.g&g2:| t50 q8<c2&¥75c2  :|

/-----------------------------------------------------
/     Gt2 (Fm1,2,3)
(t5) v9 l8
/[A]
(t5) @82 q7 |:|:o3(b,o4c)&ccc:| r1:| |:4o3(b,o4c)&ccc:|
(t5) <|:o3 q2eeq7'e<ce'q2eeq7'e4.<c4.e4.'
(t5) q2ffq7'f<cf'q2ffq7'f4.<c4.f4.':| q7o4|:8'cg<c':|
/[B]
(t5) |: o4q2'cg<c'q7'cg<c'rq2'cg<c'q7'c2 g2< c2'
(t5) >q2|:'f<cf':|q7'f4<c4f4'q2|:'g<dg':|q7'g4<d4g4'
(t5) o4q2'cg<c'q7'cg<c'rq2'cg<c'q7'c4.g4.<c4.'q2'cg<c'
(t5) |1 o3q7'f4.<c4.f4.''g2<d2g2'&'g<dg' :|
(t5) |2 o3q7'f4.<c4.f4.''g2<d2g2'|:'g16<d16g16':|:|
(t5) 'g2..<d2..g2..'|:'g16<d16g16b16':|
(t5) 'g2..<d2..g2..b2..'|:'g16<d16g16b16':|
(t5) 'g1<d1g1b1' 'd#4>g#4''f4>a4''f#4>a#4''g#4>b4'
/[C]
(t5) @24 v11 |:3o5 @d1'c4.e4.',26g&g2@d0
(t5) @d1>'b4.<d4.',26<g&g2@d0 @d1>'a4.<c4.'<g&g2@d0
(t5) @d1>'b4.<d4.',26<g&g2@d0:| o5 @d1'c4.e4.',26g&g2@d0
(t5) @d1>'b4.<d4.',26<g&g2@d0 @d1>'a4.<c4.'<g&g2@d0
(t5) @d1>'b4.<d4.',26&g4@d0 @82 v9 o4 q8(g4.c)
(t5) 'c1g1<c1',0 o3 q7 r|:7'g<dg':|
/[D]
(t5) |:  |:o3 q2aaq7'a<ea'q2aaq7'a4.<e4.a4.'
(t5) q2ggq7'g<dg'q2ggq7'g4.<d4.g4.'
(t5) |1q2aaq7'a<ea'q2aaq7'a4.<e4.a4.'
(t5) q2ffq7'f4<c4f4'q2ggq7'g4<d4g4':|
(t5) |2 q2ffq7'f<cf'q2ffq7'f4.<c4.f4.'
(t5) q2ggq7'g4<d4g4''f4<c4f4''g4<d4g4':|
/[E]
(t5) |: o4q2'cg<c'q7'cg<c'rq2'cg<c'q7'c2g2<c2'
(t5) >q2|:'f<cf':|q7'f4<c4f4'q2|:'g<dg':|q7'g4<d4g4'
(t5) o4q2'cg<c'q7'cg<c'rq2'cg<c'q7'c4.g4.<c4.'q2'cg<c'
(t5) o3q7'f4.<c4.f4.''g2<d2g2'&'g<dg' :|
(t5) o3 |:q2fff<q7'cf'&'c2f2' >q2eeeq7'b<e'&'b2<e2'
(t5) |1q2dddq7'a<d'&'a2<d2'
(t5) 'g<dg'rq2'g<dg'q7'g<dg'&'g2<d2g2':|
(t5) |2q2ffq7'f<cf'q2ffq7'f4.<c4.f4.'
(t5) q2ggq7'g<dg'q2ggq7'g4.<d4.g4.':|
/[E1]
(t5) |1  |:16'c<gc':|
(t5) v11|:|:o3(b,o4c)&cc>(b,o4c)&cccc:|r1:|v9
/[C']
(t5) @24 v11    o5 @d1'c4.e4.',26g&g2@d0
(t5) @d1>'b4.<d4.',26<g&g2@d0 @d1>'a4.<c4.'<g&g2@d0
(t5) @d1>'b4.<d4.',26<g&g2@d0 o5 @d1'c4.e4.',26g&g2@d0
(t5) @d1>'b4.<d4.',26<g&g2@d0 @d1>'a4.<c4.'<g&g2@d0
(t5) @d1>'b4.<d4.',26&g4@d0 @82 v9 o4 q8(g4.c)
(t5) 'c1g1<c1',0 o3 q7 r|:7'g<dg':|  :|
/[E2]
(t5) |2  o4q7|:|:'cg<c':|r4 >|:'g<dg':|r4
(t5)  |:'f<cf':|r'g<dg'&
(t5) |1'g2<d2g2':||2r'g<dg'r'e->a-'&
/[F]
(t5) 'e->a-'q1@82|:15'e->a-':| |:16'cg':| |:16'e->a-':|
(t5) |:16'f>b':| |:16'e->a-':| |:16'cg':| |:16'e-b-':|
(t5) |:16'd>g':| q7|:o4|:'cg<c':|ro3|:'g<dg':|
(t5) r'a<ea''a<ea'& 'a1<e1a1':||:24'g<dg':|<(c>babag)1r2
/[G]
(t5) o4q2'cg<c'q7'cg<c'rq2'cg<c'q7'c2g2<c2'
(t5) >q2|:'f<cf':|q7'f4<c4f4'q2|:'g<dg':|q7'g4<d4g4'
(t5) o4q2'cg<c'q7'cg<c'rq2'cg<c'q7'c4.g4.<c4.'q2'cg<c'
(t5) o3q7'f4.<c4.f4.''g2<d2g2'&'g<dg'
/[E]copy
(t5) |: o4q2'cg<c'q7'cg<c'rq2'cg<c'q7'c2 g2< c2'
(t5) >q2|:'f<cf':|q7'f4<c4f4'q2|:'g<dg':|q7'g4<d4g4'
(t5) o4q2'cg<c'q7'cg<c'rq2'cg<c'q7'c4.g4.<c4.'q2'cg<c'
(t5) o3q7'f4.<c4.f4.''g2<d2g2'&'g<dg' :|
(t5) o3 |:q2fff<q7'cf'&'c2f2' >q2eeeq7'b<e'&'b2<e2'
(t5) |1q2dddq7'a<d'&'a2<d2'
(t5) 'g<dg'rq2'g<dg'q7'g<dg'&'g2<d2g2':|
(t5) |2q2ffq7'f<cf'q2ffq7'f4.<c4.f4.'
(t5) q2ggq7'g<dg'q2ggq7'g4.<d4.g4.':|
/[E3]
(t5) o4|:8'c<gc':| r>|:7'g<dg':|
/[H]
(t5) |: o4q2'cg<c'q7'cg<c'rq2'cg<c'q7'c2g2<c2'
(t5) >q2|:'f<cf':|q7'f4<c4f4'q2|:'g<dg':|q7'g4<d4g4'
(t5) o4q2'cg<c'q7'cg<c'rq2'cg<c'q7'c4.g4.<c4.'q2'cg<c'
(t5) o3q7'f4.<c4.f4.''g2<d2g2'&'g<dg' :|
/[I]
(t5) o4|:64'c<c'r:| q8'c2<c2'&¥75'c2<c2' :|

/-----------------------------------------------------
/     Gt3
(t6) v11 l8 q8
/[A]
(t6) @82 o6 c2.>g4 f4.e4.d&(d,o6c)& c2.>g4 f4.g&g2 c1
(t6) >(b,o5e)&e2.. (cg)&g4(cb-)&(b-4a)&a4 (a16f)&f2...
(t6) o3(b-4,o4c)&(c4d-)&(d-4e)&e4 a1 (a,o5c)&q7c2.. q8
/[B,C]
(t6) |:30r1:|
/[D,E]
(t6) |:  |:24r1:|
/[E1,C']
(t6) |1  |:3r1:| @82 v11 o5 p2 c2.>g4 f4.e4.d4 r1
(t6) <c2.>g4 f4.g&g2 p3 |:10r1:|  :|
/[E2]
(t6) |2  r16 |:3r1:| o6 v9 q7 r2r4.c&
/[F]
(t6) @82l16(cd)&(de-)&(e-d)&(dc)&c>(b-g)&(ga-)
(t6) (a-b-)(b-g)&(ga)&(ag)&gfgl8(f16g)&g16
/このへんは(t4)と同じです
(t6) g2&(gf)&ff.f16&
(t6) f<(e-,o7e-)&e-4>(e-,o7e-)&e-4&(e-,o6e-)>
(t6) (e-2.,o4e-)<r(a16,o6f)&(f16g)&gb-f|:3(f16g)&g16:|ff
(t6) &(fg)&ga-(fg)&gg(e-f)&f e-(e-d)&d(e-16d)&d16cdc>b-
(t6) (g16f)&f16e-(c4,o4c)r>b-<ce-
(t6) (c16g)&g16b-<cd4.>>q2g4q7
(t6) l16|:o2(b-,o3c)&c:|(e-f)&fe-8e-(fg)&gfggb-g
(t6) (b-,o4c)&cc>b-<ce-ce-|:(e-f)&f:|g8(b-g)&g
(t6) l8b-<(e-16f)&f.(e-16f)&f16&(f4,o4c)>b-b-
(t6) <e-ce-f(f16g)&(g16f)&(f16e-)&e-16(fg)&g
(t6) e-(fg)&gf(f16g)&g16b-r>g <c>b-gb-4<c>b-4
(t6) gd4f(g2d)& (d1,o2g) r1 <(f1a)& a1

(t6) |:3r1:| r2...  :|

/-----------------------------------------------------
/     Kb
(t7) v7 l8 q8
/[A,B,C]
(t7) @2 |:4r1:| o3 'e1g1'&'e1g1' 'e1g1' 'f1a1' 'g1b-1'
(t7) 'f1a1' q7'g1<c1'q8 |:30r1:|
/[D,E]
(t7) |:  |:24r1:|
/[E1,C']
(t7) |1  |:18r1:|  :|
/[E2]
(t7) |2  |:3r1:| @2 o3 r2r4.e-&
/[F]
(t7) @2 e-1 d1 c1& c1 e-1 d1 c1 >b-1 <e-1 d1 c1& c1 e-1
(t7) d1 g1& g1 |:8r1:| r2  :|

/-----------------------------------------------------
/     Ba (Fm8)
(t8) t187 @28 v12 l8 q7
/[A]
(t8) o2 |:|:(b,o3c)&ccc>:| r1:| |:4(b,o3c)&ccc>:|
(t8) |:eeeeeeee ffffffff:| <cccccccc
/[B]
(t8) |: o3 cccccccc >ffffgggg <cccccccc |1 >fffg4ggg :|
(t8) |2 o2 fffg4gg4 |:3gggg4ggg:| g#4a4a#4b4:|
/[C]
(t8) |: |1 |:7r1:| o3 r2v11(a2,o2a)v12 :| |2 |:o3
(t8) cccccccc >gggggggg ffffffff gggggggg:|:|
(t8) <cccccccc >gggggggg
/[D]
(t8) |:  o2 aaaaaaaa gggggggg aaaaaaaa ffffgggg aaaaaaaa
(t8) gggggggg ffffffff ggggf4g4
/[E]
(t8) |:o3 cccccccc >ffffgggg <ceff#gfed >fefg4gg4:|
(t8) ffff4fff eeee4eee <dddd4ddd >gggg4ggg
(t8) ffff4fff eeee4eee  ffff4fff  gggggggg
/[E1]
(t8) |1o3|:cccccccc:| |:|:>(b,o3c)&cc>(b,o3c)&cccc:|r1:|
/[C']
```

▶ 今年の夏も郵便局でアルバイト。今回はMIDIを一式揃えるぞ。それにしても高校３年生がこんなことをしていていいのだろうか。　宇野　和孝(18)茨城県

```
(t8) |:o3 cccccccc >gggggggg ffffffff gggggggg:|
(t8) <cccccccc >gggggggg  :|
/[E2]
(t8) |2  o3 ccr4>ggr4 ffrf&(f,o3e)&(e4.,o2g) <ccr4>ggr4
(t8) ffrgrgra-&  :|
/[F]
(t8) |:a-a-a-a-a-a-a-a-:| <|:cccccccc:|>
(t8) |:a-a-a-a-a-a-a-a-:| b-b-b-b-b-b-b-<(cd)&dfgb-d4c4>
(t8) |:a-a-a-a-a-a-a-a-:| <|:cccccccc:|
(t8) |:e-e-e-e-e-e-e-e-:| >|:gggggggg:|
(t8) <ccr>ggraa& a1 <ccr>ggraa& a2&(a,o3a)&(a4.,o3c)
(t8) >|:gggg4ggg:| <gggggg4 {c>babag}1 r2
/[G]
(t8) o3 cccccccc >ffffgggg <cccccccc >fffg4ggg
/[E]copy
(t8) |:o3 cccccccc >ffffgggg <ceff#gfed >fefg4gg4:|
(t8) ffff4fff eeee4eee <dddd4ddd >gggg4ggg
(t8) ffff4fff eeee4eee ffff4fff gggggggg
/[E3]
(t8) o3 cccccccc r>ggggggg
/[H]
(t8) |: o3 cccccccc >ffffgggg
(t8) <cccc&v11(c,o4c)&(c4.,o3c)v12 >fff4ggg4 :|
/[I]
(t8) |:4o3cccccccc>ffffgggg<ccccccccffffeeed:|q8c2&¥75c2

/------------------------------------------------------
/     Dr (ADPCM)
(t9) l8 p3 q8
/[A]
(t9) o1 g4d4cc#d4 c4d4f#c#dc# g4d4cc#d4 cc#dc#dd#c#d
(t9) g4d4cc#d4 |:c4d4cc#d4:| c4d4cc#dc# |:c4d4cc#d4:|
(t9) dc#dc#dd#c#d
/[B]
(t9) o1 |: g4f4<e4>f4 |:e4f4<e4>f4:| |1 e4<f4g#4f4> :|
(t9) |2 e4<f4g#4f4> |:g4f4<g#4>f4:| g4f4<g#4f4 a4f4f4f4>
/[C]
(t9) o1 |: g4d4cc#d4 c4d4f#c#d4 |:c4d4cc#d4 c4d4f#c#d4:|
(t9) c4d4cc#d4 ¶1 c4d4cc#dc# :|
(t9) |2 c4d4cc#d4 g4d4f#c#d4 cc#dc#dc#dc#
/[D]
(t9) |:  o1 g4d4cc#d4 c4d4f#c#d4 c4d4cc#d4 c4dc#c4d4
(t9) c4d4cc#d4 c4d4f#c#f4& g4d4f#c#f4
(t9) <<<{e#e#e#}4{d#d#d#}4>>a4a4>
/[E]
(t9) o3>>g4<<d4cc#d4 c4d4cc#dd# c4d4cc#d4c4dc#>>b<<c#dd#
(t9)   >>g4<<d4cc#d4 c4d4cc#dd# c4d4cc#d4c4dc#>>b<<c#dc#
(t9) o1g4dc#f#4d4 c4dc#f#c#d4 |:c4dc#f#4d4 c4dc#f#c#d4:|
(t9) c4dc#f#4d4 c4dc#f#c#dc#
/[E1]
(t9) |1  o1g4d4c4dc# c4d4cc#d4 o3>>g4<<d4cc#d4 c4d4cc#d4
(t9) cc#dc#cc#dc# |:c4d4cc#d4:|
(t9) c#<ccccccc>
/[C']
(t9) o1 g4d4cc#d4 c4d4f#c#d4 |:c4d4cc#d4 c4d4f#c#d4:|
(t9) c4d4cc#d4 c4d4cc#d4 g4d4f#c#d4 cc#dc#dc#dc#  :|
/[E2]
(t9) |2  o1 gc#d#4gc#d#4 gc#d#c#rc#<<<c4>>> gc#d#4gc#d#4
(t9) gc#d#ererg&
/[F]
(t9) gc#f#4cc#f#4 |:3c4d4cc#d4:| g4d4cc#d4 |:c4d4cc#d4:|
(t9) cc#d4cc#dc# g4d4cc#d4 |:c4d4cc#d4:| c4d4cc#a4
(t9) g4d4cc#d4 |:c4d4cc#d4:| cc#dc#f#c#d4 d#d#rd#d#rd#d#
(t9) r<<<e#16e#16d#16d#16d#16d#16c#16c#16c#16>>>c#16d#4
(t9) d#d#rd#d#rd#d# r2.<<<c4>>> g4d4c4d4 c4d4cc#dc#
(t9) c4d4cc#d4 {cd#d#d#d#d#}1 r2
/[G]
(t9) o3>>g4<<d4cc#d4 c4d4cc#dd# c4d4cc#d4c4dc#>>b<<c#dd#
/[E]copy
(t9) o3>>g4<<d4cc#d4 c4d4cc#dd# c4d4cc#d4c4dc#>>b<<c#dd#
(t9)   >>g4<<d4cc#d4 c4d4cc#dd# c4d4cc#d4c4dc#>>b<<c#dc#
(t9) o1g4dc#f#4d4 c4dc#f#c#d4 |:c4dc#f#4d4 c4dc#f#c#d4:|
(t9) c4dc#f#4d4 c4dc#f#c#dc#
/[E3]
(t9) o1 g4d4cc#d4 c<<<eeeeeee>>>
/[H]
(t9) o3 |: >>g4<<d4cc#d4 |:3c4d4cc#d4:| :|
/[I]
(t9) o3 |:4 |:4c4d4cc#d4:| :|
(t9) o1 t50 g1  :|

/------------------------------------------------------

(p)            /play all tracks
/(p4,5,6,7,8,9) /karaoke
/(p4,5,6)      /Gt & Kb
/(p8,9)          /Ba & Dr
```

リスト2　恋をしようよ Yeah! Yeah! コンフィグファイル

```
/ 恋をしようよ Yeah! Yeah! ADPCM設定ファイル
.o1c# =KICK1.pcm,v45
.o1d# =SHPS.pcm,v65

.o1f# =HO2.pcm,v70
.o1g# =HO1.pcm,v70
.o1a# =CRSH1.pcm,v50
.o1b =RIDE1.pcm

.o4c# =POWT1.pcm,v30
.o4d# =POWT2.pcm,v30
.o4e# =POWT3.pcm,v30

.o1c  =.o1c#,m30
.o1d  =.o1d#,m30

.o1e  =.o1c#,m32
.o1f  =.o1d#,m32

.o1g  =.o1c#,m34
.o1a  =.o1d#,m34

.o3c# =.o1c#
.o3d# =.o1d#
.o3c  =.o3c#,m35
.o3d  =.o3d#,m35

.o4c  =.o4c#,m27
.o4e  =.o4e#,m27

.o2e  =.o1e,m25,d1300
.o2f  =.o1f,m25,d1300
.o2g# =.o1g#,m25,d1300

.o2g  =.o1g,m25,d1300
.o2a  =.o1a,m25,d1300

.erase 34
```

リスト3　恋をしようよ Yeah! Yeah! カウンタ表示

```
/ 恋をしようよ Yeah! Yeah!  ステップカウント

/ 1:000088E0 00000000   2:000088E0 00000000   3:000088E0 00000000   4:000088E0 00000000
/ 5:000088E0 00000000   6:00006540 00000000   7:000065A0 00000000   8:014D88E0 00000000
/ 9:000088E0 00000000
```

リスト4　ゆめいっぱい

日本音楽著作権協会(出)許諾第9270994-201号

```
  10 /*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*
  20 /*          FOR  X68000+SC55                        /*
  30 /*                  ーゆめ　いっぱいー              /*
  40 /*        ーちびまる子ちゃんTVアニメバージョンー    /*
  50 /*             <作曲/編曲        織田　哲郎>        /*
  60 /*                                                  /*
  70 /*      PROGRAMED      BY      田所　広行           /*
  80 /*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*
  90 /*
 100 m_init():m_ch("MIDI"):key 3,"m_stop()"+chr$(13):sc55_print
("Chibi Maruko Chan")
 110 dim char rs(3)={&H40,0,&H7F,0}
 120 m_roland(&H10,&H42,rs)  /*SC55を工場出荷状態に初期化
 130 for i=1 to 16:m_alloc(i,1500):m_assign(i,i):next
 140 str a[256],b[256],c[256],d[256],e[256],f[256]
 150 str x[12]="@i$41,17,$42"
 160 dim char rv(6)={4,4,0,&H70,&H40,0,0}:sc55_reverb(rv)   /*
リバーブ設定
 170 dim char ch(7)={2,0,&H6E,8,&H50,3,13,0}:sc55_chorus(ch)/*
コーラス設定
 180 dim char dr(3)={&H40,&H1A,&H15,&H2}
 190 m_roland(&H10,&H42,dr)      /*PART11をドラム2に設定
 200 dim char dm(3)={&H40,&H1B,&H15,&H1}
 210 m_roland(&H10,&H42,dm)      /*PART12をドラム1 に設定
 220 dim char dn(3)={&H40,&H1C,&H15,&H1}
 230 m_roland(&H10,&H42,dn)      /*PART13をドラム1 に設定
 240 for i=1 to 16:m_trk(i,x+"r2"):next
 250 dim char pr(15)={1,2,4,2,1,1,2,1,0,1,2,1,2,2,1,1}:sc55_v_r
eserve(pr) /*パーシャルリザーブ
 260 dim char ks(3)={&H40,&H1A,&H16,&H3E}:m_roland(&H10,&H42,ks
) /*パート11をキーシフト
 270 /*
 280 /*
 290 /*  MML  設定
 300 /*
 310 /*
 320 /*シンセサイザー1
 330 a="l16 z127,120,115,120,127 r4dc+>b<c+d2& d4f+edef+2 z q7g
2l4cg gf+ed gd+ga+ afa<cq8
 340 b="l8>_15b-1a1g1g2a4<c4>b-1a1g2e-.f.ga2a.g.ab-1g1 _10g-4&f
4&e-4&d-4&c2f2&
 350 c="_10f1&_10f2_10&f2l1rr >b-gcf ~30ddgg cf~10b-2a2~15b-2<c
```

```
2>_25e-1f1b-1g1r1r1r1r2.
  360 d="¯25d16f16<d16f16_15l8b-1a1g1g2a4<c4>b-1a1g2e-.f.ga2a.g.
ab-1b-2
  370 m_trk(1,"@y1,$63,$3d@51o5v9@e70,90"+a):m_trk(15,"@y1,$20,$
35v9@49o5@p10@e60,60"+a)
  380 m_trk(1,b):m_trk(15,b)
  390 m_trk(1,c):m_trk(15,c)
  400 m_trk(1,d):m_trk(15,d)
  410 /*シンセサイザー2
  420 a="l16 z127,120,115,120,127r4bagab2& b4<dc>b<cd2 z q7e2>l4
a<e dc+>ba <d+>a+<d+g fcfaq8
  430 b="_15l8f1f1e-1e-2f4g4f1f1e-2>b-.<c.de2e.e-.ff1>b-1 >_8'b-
4<b-'&'a4<a'&'g4<g'&'g-4<g-'&'f2<f'<'f2<c'&
  440 c="_10'f1<c'&_10'f2<c'_10&'f2<c'l1rrrrrr ¯30'g-a''g-a''b-<
d''b-<d''e-g''a<c'¯10'df2''d-e2'¯10'dfa-2''egb-2'_17'gb-1''a<ce-
1''<df1''b-<d1'r1r1r1r1
  450 d="¯5l8<f1f1e-1e-2f4g4f1f1e-2>b-.<c.de2e.d+.ff1f2
  460 m_trk(2,"@y1,$63,$3d@51o4v7@e70,90"+a):m_trk(16,"@y1,$20,$
35v7@49o4@p110@e60,60"+a)
  470 m_trk(2,b):m_trk(16,b)
  480 m_trk(2,c):m_trk(16,c)
  490 m_trk(2,d):m_trk(16,d)
  500 /*シンセサイザー3
  510 a="@e80,20@49v10l1o2 'g<dg<d''b<f+b<f+'<'cg<cg''dg<dg''d+g
<d+g4'>'a+<d+a+<d+4'<'d+g<d+g4''ga+<ga+4''fa<fa4''cf<cf4''fa<fa4
''a<ca<c4'
  520 b="_10>'b-<df<b-'<'dfa<d''e-gb-<e-''fa<ce-'>'b-<df<b-'<'df
a<d''e-gb-<e-''fa<ce-'>'b-<df<b-''gb-<dg'<'g-b-<d-''fa<c'
  530 c="@7v9@e90,10o3@y1,$20,$3bl4|:|:i127,0'b-<df.',4:|'b-<df'
 |:'gb-<d.':|'gb-<d' |:'ce-gb-.':|'ce-gb-' |:'fa<ce-.':|'fa<ce-'
:||:|:'dg-a<c.':|'dg-a<c':||:|:'gb-<d.':|'gb-<d':|
  540 d="|:'ce-gb-.':|'ce-gb-'|:'fa<ce-.':|'fa<ce-'|:'b-<df.':|'
b-<df'|:'b-<dfa-.':|'b-<dfa-' @48o4l16 @y1,$20,$40 @y1,$64,$50@e
110,20v10|:i0,0e-gb-<e-ge->b-g:||:i0,0fa<ce-fe-c>a:||:i0,0b-<dfb
-<d>b-fd>:||:gb-<dgb-gd>b-:|
  550 e="@e120,90@31@y1,$21,$4a@y1,$20,$33o3l8 r8v7|:4'c>c',0:|r
2|:4'c>c':|q7r4.@b0,-4000,70'f>f2'@b0,-4000,50'f>f4.'@b0'f>f16''
f>f16'|:4'f>f4':|
  560 f="@y1,$20,$40@y1,$21,$40q8@e80,20@49v9l1o2'b-<df<b-'<'dfa
<d''e-gb-<e-''fa<ce-'>'b-<df<b-'<'dfa<d''e-gb-<e-''fa<ce-'>'b-<d
f<b-''b-<df<b-2'
  570 m_trk(3,a)
  580 m_trk(3,b)
  590 m_trk(3,c)
  600 m_trk(3,d)
  610 m_trk(3,e)
  620 m_trk(3,f)
  630 /*シンセサイザー4
  640 a="@e120,80 @y1,$20,$3a @y1,$64,$55 @11o6v5l4 d1r1r1<gf+ed
v09 q5gd+ga+ afa<c>b-1a1g1afa<c>b-1a1g2e-8.f8.g8a2a8.g8.a8b-1_15
g1g-fe-d-c1|:14r1:|¯15b-2a2a-1 g1a1b-2b-2g1 r1r1r1r1
  650 b="b-1a1g1afa<c>b-1a1g2e-8.f8.g8a2a8.g8.a8b-1
  660 m_trk(4,a)
  670 m_trk(4,b)
  680 /*サクソフォン
  690 a="@y1,$20,$39@y1,8,$4c@y1,9,$4bl16r1r2...._30c32&g4r2. ¯2
0f32&g8..e32&f+8..d32&e.@h20@m100@b0,-2000,44d4.@b@mc.cd8e8@m90e
4@md8c8@h50@m100d&f2..."
  700 b="|:8r1:|@h100r2d4. f&e&d+@k64d&@k0d1^8 f+4&f8&d+8 c+4&f2
^8 f4&d+8 d1
  710 m_trk(5,"@66o4v14@e100,20"+a):m_trk(6,"@b50@66o4v14@e100,2
0"+a)
  720 m_trk(5,b):m_trk(6,"@b50"+b)
  730 /*シンセベース
  740 a="@e60,10@y1,$20,$30@38v12o1l8q6 |:4g<g>:||:4b<b>:|<|:4c<
c>:||:4d<d>:||:4d+<d+>:|f<f>f<f>fcdf
  750 b="_10o2q8>|:b-fgb-4fgb- <d>ab<d4>ab<d e->b-<ce-4>b-<ce- f
cdf|&fcdf>:|f.f.f |:4b-<b->:||:4g<g>:||:4g-<g->:||:4f<f>:|
  760 c="¯15|:l4b-b-b-b-8b-8 gggg8g8 cccc8c8 ffg8f8b-8f8:|<dddd8
d8 dddd8d8 >gggg8g8 gggg8g8 cccc8c8 ffff8f8 b-b-b-b-8b-8 b-b-b-b
-8b-8
  770 d="e-.e-.e-8e-8 f.@b0,-2000,0f8&@b-2000,0f8&@b0f8f8f8 b-.b
-.b-8b-8 g.gd8e8g8 r8c8c8c8c8d8e8 f8c8c8c8c8d8e8 f2f2fff8c8f
8c8
  780 e="_10o1l8|:b-fgb-4fgb- <d>ab<d4>ab<d e->b-<ce-4>b-<ce- fc
df|&fcdf>:|f.f.f b-1 b-1
  790 m_trk(7,a)
  800 m_trk(7,b)
  810 m_trk(7,c)
  820 m_trk(7,d)
  830 m_trk(7,e)
  840 /*ドラムス1(BassDrum)
  850 a="@r1@e100,10v09@1l4o2k-1r8|:12cccc:|cccc8cc16c16c8c6c6c6
 r8|:3cccc:|cccc8 ¯10|:8c2..c8c8r2..:||:c.c.c c.cc8c:|c2c2c2c2cc
ccccccr8|:6cccc:|cccc8cc16c16c8c6c6c6r8cccc
  860 m_trk(10,a)
  870 /*ドラムス2(SnareDrum&tamburin)
  880 a="@r1@e120,5v10@p80@49 l4o2|:20d:|dd*42d32d8d8d16d16d8 |:
7'd<f+''d<f+''d<f+''d<f+':|'d<f+'d16d16d8'd6<f+''d6<f+''d6<f+' |
:14'd<f+':|'d<f+8'|:5d16:|r16 |:32rd:|
  890 b="|:r2.'d<f+'r2.d:|¯15r8d8d8d8d8r2d8d8d8d8r.dddddd8d16d1
6d16d16d8_15
  900 c="|:7'd<f+''d<f+''d<f+''d<f+':|'d<f+'d16d16d8'd6<f+''d6<f
+''d6<f+' 'd<f+''d<f+''d<f+8'|:5d16:|
  910 m_trk(11,a)
  920 m_trk(11,b)
  930 m_trk(11,c)
  940 /*パーカッション(ﾊﾝﾄﾞｸﾗｯﾌﾟ)
  950 a="@r1@e50,0v11@1l8o2 |:5d+4d+4d+d+d+4:| d+4d+*42d+32d+d+d
+16d+16d+|:7d+4d+4d+d+d+4:|d+4r16r16rd+6d+6d+6|:4d+4d+4d+d+d+4:|
|:16r1:||:8r1:||:7d+4d+4d+d+d+4:|d+4r16r16rd+6d+6d+6d+4d+4d+d+d+
4d+1
  960 m_trk(12,a)
  970 /*ドラムス3(hi-hat & Cymbal)
  980 a="@r1@e50,30v8@1l8o3 c+>|:47f+:|<<<v6@p40l16|:120a+:|o3c+
6c+6c+6o5|:64a+:|o2l8|:128f+:||:32f+:|rf+f+f+f+r2f+f+f+f+r4.a+f+
f+f+f+f+f+f+f+f+f+f+a+f+f+o5l16|:120a+:|o3c+6c+6c+6o5|:16a+:|o
3c+1
  990 m_trk(13,a)
 1000 /*merody
 1010 a="@e80,20v16o5@h50@m30q7l8 |:8r8i127,0:|r1@48r1r1r1r1_15
 1020 b="¯15ffffddf>a4a4a<d4fr ggggd+d+gg4ag4f4d+r ffffddfd4d4df
4ar ggggd+.f.ga4.ra^16g^16aa+1&a+2r2r1r1>
 1030 c="_15r4a+4aa+4<c& c4>a+4aa+4<d &d2d+4dd &dc2.r> r4a+a+a+a
+a+a+ <c4>a+a+aa+4<g &g4g4g4ff& f2.dd+
 1040 d="d4d+d2dd4d+4d4c4c4.c>a+4aa+2.g4a+<d4cc4.c>a+aa4ag4aa+1&
a+2..b-4
 1050 e="<g2.b-4a4a4ga4ab-4b-4b-a4gd2r dddc2rdddc4.r>'fb-4'<'gd'
'fc''gd''cf2.'&'cf1'r
 1060 m_trk(14,a)
 1070 m_trk(14,b)
 1080 m_trk(14,c)
 1090 m_trk(14,d)
 1100 m_trk(14,e)
 1110 m_trk(14,b)
 1120 m_tempo(140):m_play()
```

(善)のゲームミュージックでバビンチョ

セガ「シャドーダンサー」の基板を買った。このゲーム，やたら難しくてゲームセンターでやってた頃はステージ１さえクリアできなかったんだが，これで思う存分遊べます。この基板，ナムコのシステム２みたいになんとかステレオに出力できないかしらん。どーもCRTから聴こえるこもった音では迫力に欠けるので。情報待つ。

●ダライアス/TAITO ZUNTATA
CD:PCCB-00093
ポニーキャニオン　1,500円(税込)　8/21発売

以前に紹介した「OUTRUN」「忍者ウォーリアーズ」同様の復刻版シリーズ。オリジナルサウンドにアレンジ２曲を収録。しかし，あの頃のタイトーって「忍者ウォーリアーズ」にしろ，この「ダライアス」にしろ，他社とは違った独自の音楽センスで光ってましたねぇ。

「ダライアス」では，「COSMIC AIR WAY」は私は名曲だと思うな。YMOを彷彿させるようなちょっと日本風味の味つけのメロディに，わが道を行くリズム。メロディもなんか裏返ったようなどぎついFMシンセ，しびれるなー。ダライアスというとみんな「CAPTAIN NEO」に目がいきがちだけど私は「COSMIC～」を推すな。

さて，アレンジバージョンはなんとなくこの間聴いたYMOのリミックスアルバム「HI-TECH/NO CRIME」(ALFA:ALCA-323)を思い出させた。

お勧め度　8

●スーパーファミコン・ゲームミュージック　サイバリオン
CD:TOCT-6617
東芝EMI　3,000円(税込)　8/12発売

スーパーファミコン(以下SFC)の音源てやつは，スゴイスゴイと初めはいわれていたけど，関係者から話を聞くといろいろと問題・制約があるようで。んで，各社独自の技術でそれらを解決しているんだね。このSFC版サイバリオンも，そういった問題・制約をちっとも感じさせない，かなりの完成度だ。AC版とはまったく違った，全曲SFC版のための書き下ろし曲が収録されている。

私はSFCも持ってないし，SFC版サイバリオンも遊んだことはないのだが，そんな私でもこのアルバムはすっかり気に入ってしまった。２トラック目の「戦い～勇者朝日に向かう」ってやつがいいね，ちょっとダンサンブルで。リズムを補間する感じのベースライン，単純なピアノのバックに乗って展開するアドリブ調メロディ。

ちなみに音楽監修はすぎやまこういち氏だそうです。

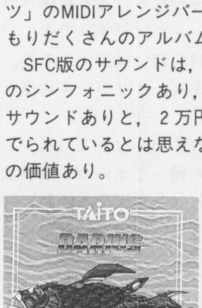
お勧め度　9

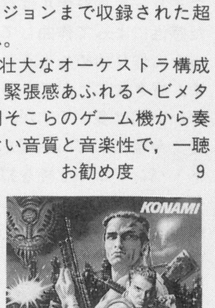
●魂斗羅スピリッツ/コナミ矩形波倶楽部
CD:KICA-7604
キングレコード　2,800円(税込)　8/21発売

SFC版の「魂斗羅スピリッツ」のオリジナルゲームサウンド全曲と，アーケード版，ファミコン版，MSX2版，ゲームボーイ版の「魂斗羅」「スーパー魂斗羅」のダイジェスト，さらに「魂斗羅スピリッツ」のMIDIアレンジバージョンまで収録された超もりだくさんのアルバム。

SFC版のサウンドは，壮大なオーケストラ構成のシンフォニックあり，緊張感あふれるヘビメタサウンドありと，２万円そこらのゲーム機から奏でられているとは思えない音質と音楽性で，一聴の価値あり。

お勧め度　9





善バビスペシャル

対談!! GMコンポーザー

第4回 S.S.T.BAND

今月はセガ・サウンド・チーム，「S.S.T.BAND」の皆さんのお話をうかがうことができました。

松前公高(以下松)：キーボードとコンピュータをやってます，松前です。

並木晃一(以下並)：ギターとMCとリーダーをやってます，並木です。

飯島文治(以下飯)：ギターとオタクをやってます，飯島です（笑)。

西川善司(以下善)：さて，FIのベルガーのテーマということでS.S.T.BANDの曲がこの間かかりましたが，ということはつまりT-SQUAREなどと肩を並べたということですかね。

飯：いや，でも身長はそんなに変わんないと……(笑)。

善：新幹線「のぞみ」のプロモーションビデオで内装かなんかを紹介しているとき，バックでギャラクシーフォースがかかってましたよ。

飯：うーむ。それでビス飛ばされちゃった日にはたまりませんねー（笑)。

S.S.T.BANDとセガ・サウンド・チーム

善：BLIND SPOTの発売以後，ずいぶんと前面にクローズアップされてきていますが，ちゃんとゲームミュージックも作られているんですか。

並：ええ。それがメインで，逆に合間にこういったバンド活動をやっています。

善：ふーむ。よかった。S.S.T.BANDというのは総勢何名くらいで構成されているのですか。

並：ええと。アーケードなどのゲームミュージックを担当しているのはだいたい20名くらいですね。LIVEやBLIND SPOTなどのようなS.S.T.BANDの活動をしているセガの社員は2名です。

善：すると，S.S.T.BANDってたしか6人構成だったと思いますが，あとの4人は？

飯：ふだんはふつうのミュージシャンです。だから私ゲームミュージックよく知らない（笑)。

(注：今回お話をうかがった3人のうち，セガの社員でかつS.S.T.BANDの方は並木さんだけです)

善：オリジナルとゲームミュージック，2つのお仕事に感覚的な違いってありますか。

並：ゲームの場合には目の前に絵がありますが，オリジナルの場合はそういうのがないですよね。つまり，オリジナルは自分のなかにある蓄積された感性によって作曲していくわけです。でも，いったん作り始めちゃうと特に違いはないですね。ま，好きなことができるぶん，オリジナルのほうがいい面は多いですが。

善：今後どんどん活躍されていくなかで，インスト・バンドという枠を打ち破ってボーカルが参加してきちゃうみたいなことはありえますかね。

松：うーん，周りに歌がうまいやつがいないっていう……（笑)。

並：いきなり，こいつのために曲が書きたい！なんていうボーカリストが現れたりしたら，たぶん……。

松：基本的に「声」というひとつの楽器としてとらえていますし。我々は特にインストというジャンルにこだわりはありません。

善：そうですか。んじゃ，ちょっと質問の方向を変えまして。せっかくOh!Xですから(笑)。ちょっと突っ込んだ質問をいくつか。セガさんは曲作りと打ち込み人というのは分かれているんですか。

並：いえいえ，うちは昔っからひとりで全部みんなやりますよ。効果音から打ち込みからね。音源ドライバなどもやりたくはないんですが，自分でやってますね。

善：ファンタジーゾーンとか，アウトランとか，ギャラクシーフォースとか，いろいろ好きなのがあるんですが……。

並：あ,ギャラクシーフォースの「Beyond the galaxy」とかは私の作品です，どーも（笑)。

善：……そこで，年々ハードがパワーアップして音源の発音チャンネル数などが拡大されてきていますね。そういった進歩が曲作りにどういった影響を与えますか。

並：発音数が増えたから，というようなことは，我々制作者側からはそれほど大きな問題ではないですね。やはり，小容量のROMを使っていいものを作るといったことが，我々技術者にとっていちばん要求されることですね。なかなか辛いところなんですけど。

善：ふーむ。というと，ただ音楽ができるというだけではダメなんですねぇ。

並：ええ，それはもう。我々は作曲家というよりは技術者ですから（笑)。

善：ゲームの企画の段階から曲を書き始めちゃうんですか，それとも画面などができてからとか？

並：企画からかかわる場合もありますし，ま，いろいろですね。音関係はすべてこちらに任せてくれる場合もありますが，なかにはドラムのフレーズまで指定してくる場合もありますよ。過去の例から見ると，全部任せてくれたほうが，いいものが上がりやすいですね。

善：ほかのライバル社のゲームミュージックとかを意識されたりしますか。

並：うーん。どんなチップ使ってるかとか音源は何かとか，そういった技術者の目で見ちゃいますね。やはり。

善：FM音源などは最近のゲームミュージックなどでは常識化していますが，ああいった音源でいかに面白い音を出してやろうかという研究はいつもされているのですか。

並：はい。それは技術者の宿命です(笑)。うちでは膨大なFM音源音色とPCM音色を1カ所に集めて管理し，それをネットワークによりアクセスできるようにしていますね。

善：おぉ。それはすごい。

音楽と音源

善：皆さん最近一目置いてるMIDI楽器などは？

松：やはりローランドのサウンドキャンバスSC-55ですかね。

並：実は我々みんな持ってるんです，あれ。よくSC-55は音が悪いなどというDTMマニアの方がいますが，どういった了見でそんなことをいうのでしょうかねぇ。我々が音楽を始めた頃と比べれば，考えられないほどいい音が揃ってると思うのですがね。結局，使う人のセンスの問題だと思うんですが。

松：楽器のスペックと音楽のよさとはほとんど無関係なんですよね。極端な話，ギター1本ピアノ1台でもいい曲なんてのはいくらでもできますし。

並：サンプリング周波数がいいだとか，SN比がいいだとか，そういったカタログ的なスペックよりは，要はその楽器の出す音が音楽的であるかということだと思うんです。もしそうでなかったら，1970年代の音楽を全部否定することになってしまうんで。ま，こういうメッセージを最近のDTMマニアの方々に送りたいですね。

松：私なんか，ピアノと茶碗と箸とラジカセで音楽作ってましたからね。

並：旅先なんかにも持っていけますし，SC-55は重宝してます。

松：SNとかアタックとかそういった面ではSC-55は弱いところがありますが，U220やDシリーズなどのいい音ばかりをセレクトしてありますから。ひとつのシンセの使い方をマスターするまでには，かなりの時間を要しますよね。でも，そんな時間があったら音楽作ってたほうがいいわけです。ですから，いい音をチョイスしてプリセットしていくということは大事なんですよね。ま，そんな意味からしてもSC-55ってのはすごいわけです。

飯：ローランドさん，なんかくれないかな。

一同：大爆笑

松：X68000にも内蔵音源がありますよね。だから，SC-55がなくてもいくらでも音楽はできますよ。音源のスペックにとらわれないで音楽してもらいたいですね。

並：そうそう，あのJACCSのCMなんか，そこらのもの叩くだけで音楽してたでしょ。

飯：あぁ，あのT2の警官みたいな白人がキャベツ切ったりするやつね。

松：シーケンサでも使いにくかったらそのシーケンサの操作体系にインスパイアされて，そのシーケンサならではの曲を作ることもできますね。

善：なるほど……。

BLIND SPOT

善：アルバム「BLIND SPOT」では，5トラック目の「SEVENTH FLIGHT」というのが好きなんですが，メンバーの方に直接なんかコメントがいただけると鑑賞の幅が一層広くなると思うので。

松：「SEVENTH FLIGHT」のアドリブ的なシンセ・ソロは，実は私ともうひとりのキーボーディストで作ってるんです。あれは半分が手弾きで半分が打ち込みで，コンピュータと人間との融合を実現しています(笑)。初めから終わりまで完全に楽譜化した曲よりも，その場その場の感覚で仕上げて

いくといった曲のほうが楽しいですね。

飯：頼まれもしないのに，勝手にソロで埋めちゃったりしたなー（笑）。

並：メンバーどうしがインスパイアされて生まれてくるフレーズは大事にしたいなと思っています ね。私の音楽の最終目標の一部でもありますね。

飯：将棋の名人戦みたいな。「うーむ。そうきましたか」みたいなね（笑）。

松：ただ，そればかりだとムチャクチャになるので，ある程度のガイドラインは必要ですけどね（笑）。

これからのS.S.T.BAND

善：あのー，海外進出とかは考えてないんですか？　ゲームミュージックというジャンルは，海外でもっとも知名度が高い和製音楽だと思いますし，セガの知名度もかなりのものですよね。

並：うーむ。イタリアでは「セガ」ってのは「オナニー」という意味の言葉らしいですね（笑）。ま，それはさておき，海外進出はぜひポニーさんの力でなんとかしていただきたいな，と（笑）。

飯：自衛隊じゃないですが，行けといわれたら行きます，我々は（笑）。

並：飛行機，嫌いだしなー。新幹線で行けたらいいんですが。

飯：他社さんが海外レコーディングとかいって盛り上がってましたな。

並：そういやカシオペアのインタビュー記事で，オーストラリアでレコーディングしたときの１日のスタジオ代が10万円程度だったってのがあった。我々が使ってるところより安いよね。

飯：海外のほうが安くあがるな……。

善：（海外進出っていったい……）では，最後に８月のGAME MUSIC FESTIVAL '92に向けての意気込みを。

松：皆さんのリクエストになるべくお応えした構成で，オリジナル，ゲームミュージックともに，ライブならではのアレンジを施してお届けする予定です。

飯：ま，お祭りなんで。笑かすことに命張りますんで（笑）。

並：年内にもう１枚アルバム出しますんで，よろしくー。

なんだかとても知的な感じのするS.S.T.BANDの皆さんでした。

というわけでゲームミュージックに，オリジナルに，ライブに，パワー全開のS.S.T.BANDをこれからも応援してあげてください。んでは。

S. S. T. B A N D ディスコグラフィー

★ギャラクシーフォース
CD:D28B0002　2,627円(税込)
1988年７月21日発売

★パワードリフト＆メガドライブ
CD:D28B0010　2,627円(税込)
1988年12月28日発売

★スーパーソニックチーム
CD:PCCB-00009　2,500円(税込)
1989年10月21日発売

★メガセレクション
CD:PCCB-00014　2,500円(税込)
1989年12月15日発売

★アフターバーナー／セガ（G.S.M. 1500シリーズ）
CD:PCCB-00032　1,500円(税込)
1990年６月21日発売

★ハイパードライブ
CD:PCCB-00035　3,200円(税込)
1990年７月21日発売

★S.S.T.BAND LIVE
CD:PCCB-00042　2,500円(税込)
1990年10月31日発売

★フォーミュラ
CD:PCCB-00059　3,200円(税込)
1991年４月21日発売

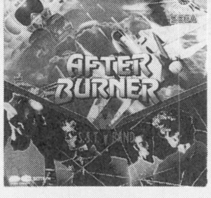
★ストライクファイター／セガ（G.S.M.1500シリーズ）
CD:PCCB-00067　1,500円(税込)
1991年８月21日発売

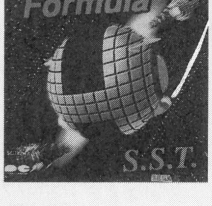
★メガセレクションII
CD:PCCB-00077　2,500円(税込)
1991年12月15日発売

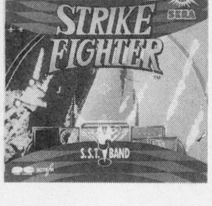
★ヴァーチャルオーディオ F-1GP ザ・エグゾースト・サウンド
CD:PCCH-00015　2,800円(税込)
1992年１月21日発売

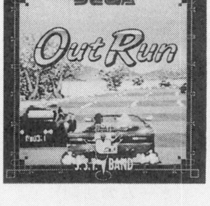
★アウトラン／セガ（G.S.M.1500シリーズ）
CD:PCCB-00081　1,500円(税込)
1992年２月21日発売

★BLIND SPOT
CD:PCCB-00085　2,800円(税込)
1992年４月29日発売









ハードウェア工作入門〔27〕

コンピュータアーキテクチャ編

デジタル論理回路を学ぶ

Misawa Kazuhiko
三沢 和彦

今月は，先月号で設計した1桁加算器を2桁に発展させ，デジタル回路で設計し直します。そこで，ソフトウェアで設計したものを，効率よくハードウェアで設計するための知識を解説していきます。

今月は8月号で設計した加算器を実際のデジタル回路で設計するために必要な事柄について説明していきたいと思います。ハードウェアに関してはまったく初めて，という読者のためにも，電子回路の基礎的な知識から説明を始めていきます。そして，実際のデジタル論理回路でよく使われている，汎用ロジックICについて説明を続けていくつもりです。

それらの説明後，先月設計した加算器について，もう一度ハードウェアの観点から考え直してみましょう。先月はあくまでも論理演算というソフトウェアの観点から設計したので，実際にハードウェアを製作するためには，もう少し検討すべき点があるのです。今月は，1桁どうしの加算をデジタル回路で実現することから始め，さらに桁上げを考えた2桁どうしの加算器まで発展させていきたいと思います。

なお，今月は予定していた以上に内容が膨らんでしまいましたので，工作実習は来月に回すことになりました。

電子回路の基礎

まず，電子回路とはどういうものかについて少し説明したいと思います。図1の豆電球に電池をつないで点灯させる回路を見てください。この回路を考えるうえで電圧と抵抗，電流という3つの要素を理解することが重要です。電圧は回路に電気を流そうとする力，抵抗は電気の通りにくさの度合い，そして電流は回路に流れている電気の量のことをいいます。これら電圧E，抵抗R，電流Iの間には，

(電圧E)＝(抵抗R)×(電流I)

という単純な関係が成り立ち，これを「オームの法則」といいます。

図1の回路で電池の電源電圧をE＝6V(ボルト)，電球の抵抗をR＝100Ω(オーム)とすると回路に流れる電流は，

I＝E÷R

で計算でき，6÷100＝0.06A(アンペア)となることがわかります。

今度は，この回路にスイッチを付けてみましょう。スイッチを切ったときには電球は消えており，スイッチを入れると電球がつきます。このときの電圧と電流の様子を調べてみましょう。図2のように電流計と電圧計とを回路につないで電球に流れる電流と電球の両端の電圧を測定します。電流はスイッチをONにしたときには0.06Aになっていますが，スイッチをOFFにしたときには0Aであることがわかります。これは，電球がつくときは電流が流れているときであることに対応しています。そして，電球の両端の電圧についても，スイッチがONのときに6V（Hレベル）で，OFFのときには0V（Lレベル）になっています。

以上の例からスイッチのON/OFFを電圧のH/Lに対応させることができるのがわかるでしょう。前回述べた論理演算では，すべての数値を1/0の2種類で表すことになります。これをデジタル論理回路で行おうとするならば，電圧レベルのH/Lの2種類で表すのが適当なところです。ところで，論理演算は基本的にAND，OR，NOTの3種類ですべて表現することができる，といいました。これにしたがって，デジタル論理回路を考えるうえでもこの3種類の基本演算を実現する回路を組むことが，最初の問題となるわけです。

では，電池とスイッチとで豆電球を点灯させる回路を使って，論理回路のAND，OR，NOTに相当する回路を作ってみましょう。図3の3つの回路を見てください。図3(1)はAND回路になっていて，スイッチ(a)とスイッチ(b)とがどちらもONになっていないと電球が点灯しません。スイッチのON/OFFをH/Lに置き換えれば，まさにAND回路の論理になっています。図3(1)の下にある論理表を参照してみてください。

次に図3(2)のOR回路は，スイッチ(a)とスイッチ(b)のどちらか一方がONになって

図1

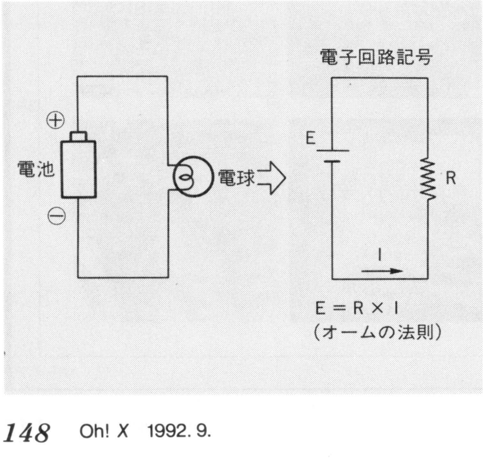
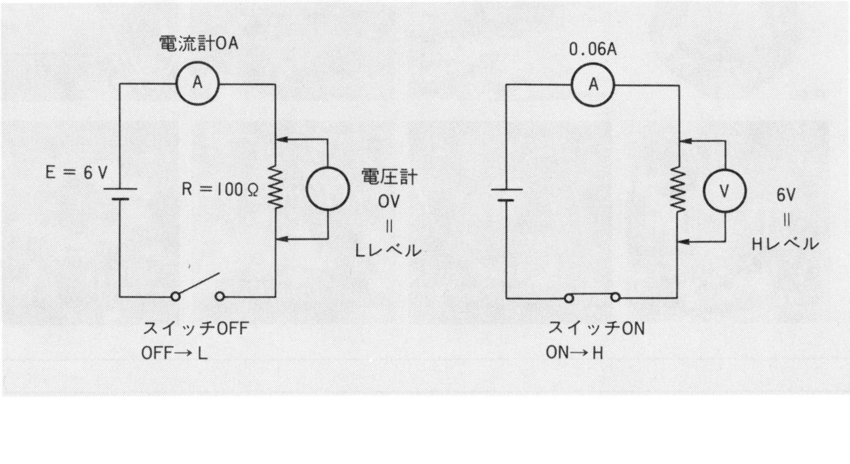


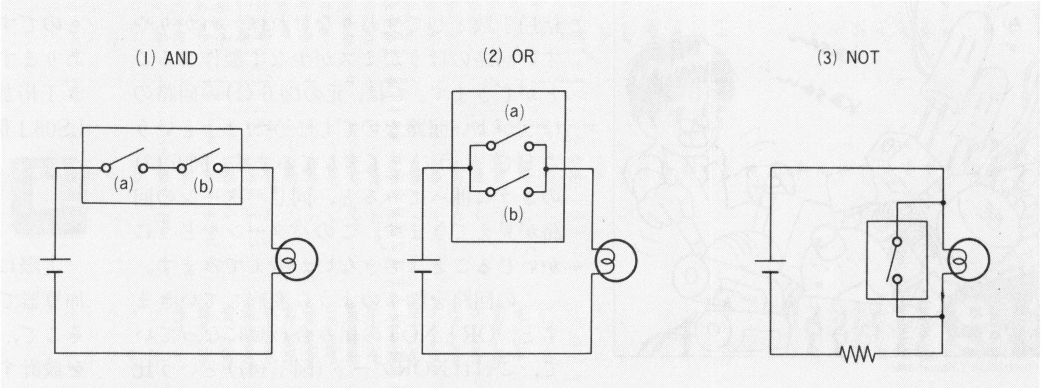
図2　スイッチのON/OFFによるH/Lレベルの区別



いさえすれば，電球は点灯します。これも論理表を見てください。なお，図3(3)の回路は余計な抵抗が入っていますが，とりあえず無視してください。この回路でスイッチがOFFの状態では，電流が電球のほうを流れるため点灯しますが，スイッチをONにすると今度は電流がスイッチのほうに流れてしまい，電流が電球のほうに流れなくなってしまうので消えてしまいます。したがって，スイッチのON/OFFと電球のON/OFFとが入れ替わっているわけです。これはまさにNOT回路に相当しています。

このように電圧のH/Lで論理演算AND，OR，NOTの1/0を表現する回路を構成することができるのです。

図3　豆電球を使ったAND，OR，NOT回路



(b)スイッチ ＼ スイッチ(a)	ON	OFF
ON	電球 ON	OFF
OFF	OFF	OFF

(b)スイッチ ＼ スイッチ(a)	ON	OFF
ON	電球 ON	ON
OFF	ON	OFF

しかしながら，いま述べたように豆電球とスイッチの回路では，かさばるうえに使い勝手もよくありません。そこで，論理演算を行う回路をパッケージしたデジタルICの出番となるわけです。

基本デジタルIC

デジタルICには大きく分けて，

1)　汎用ロジックIC
2)　専用LSI

の2つのグループで考えることができます。今後，使っていくのが1)の汎用ロジックICで，これはコンピュータの基礎となる論理回路であるAND，OR，NOTの基本回路をベースに，それらの組み合わせでできる回路をパッケージにしたものです。これにはTTL，C-MOSといういくつかのファミリに分類されています。同じファミリのなかで電気的特性が揃えられていて，接続が簡単になるようにしてあるのです。いくつかあるファミリでも，TTLのLSタイプとC-MOSのHCタイプがよく使われています。LSタイプは現在のデジタル論理回路の標準となっているシリーズで，極めて豊富な種類のICが揃っています。この連載ではこのLSシリーズをメインに使っていくつもりです。

一方，C-MOSは消費電力が小さく，電源電圧も2～6Vと幅をもたせることができるため，使い勝手がよくなっています。これまでの主流だったLSシリーズとほぼ完全に互換性があるので，従来の設計に変更を加えることなく，HCシリーズを使うことができます。しかしながら，LSシリーズのすべての品種をカバーしているわけでなく，実際，今回の加算器に使おうと思ったICもLSシリーズにはあって，HCシリーズにはありませんでした。HCシリーズは電源に電池を使うことができるので非常に魅力的ですが，今回はLSシリーズを使うことに絞ることにします。

1桁加算器の製作

先月述べたように1桁の加算器は，排他的論理和XORと呼ばれる論理回路と同じものです。XORの回路図は図4のとおりで，上の位はANDゲート1個のみ，下の位は4つのゲート回路（AND 2個，OR 1個，NOT 1個）の組み合わせになっています。

実際にデジタルICで組むときも原理的には，これらの4つのゲートの端子間を配線してやればよいことになります。ANDゲートはLS08，ORゲートはLS32，NOTゲート（インバータ）はLS04という型番のものです。しかし，この回路には少し工夫の余地があります。というのも，ひとつのパッケージには，同じゲートが複数あり，LS08，LS32では4個ずつ，LS04にいたっては6個も入っているのです。それでも，AND，OR，NOTは別々のパッケージに入っているため，最低3個のICは必要になります。実際に使うゲートは，それぞれの種類で1

図4　1桁加算器

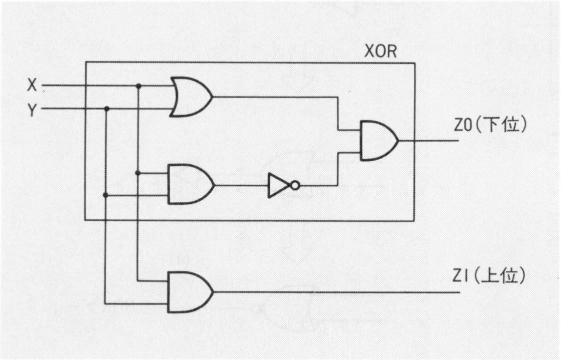
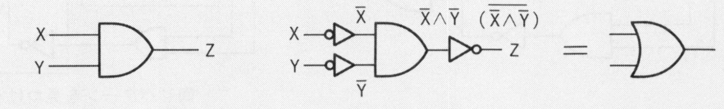


図5　ゲートの組み替え



Y ＼ X	H	L
H	H	L
L	L	L

AND回路

X Y	X̄ Ȳ	X̄∧Ȳ	$(\overline{\bar{X}\wedge\bar{Y}})$	X∨Y
L L	H H	H	L	L
L H	H L	L	H	H
H L	L H	L	H	H
H H	L L	L	H	H

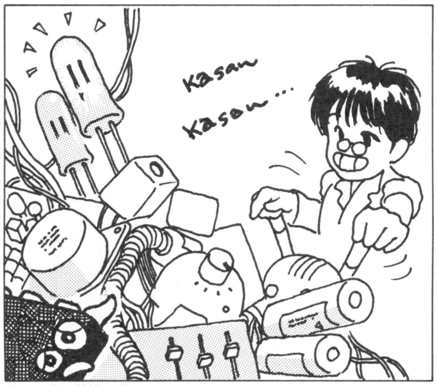


illustration:Y.Kawahara

個か2個なので，大部分のゲートは使わないままになってしまうわけです。

これは大変もったいないことです。そこで，1個しか使わないゲートをほかのゲートの組み合わせで表現することを考えます。図5を見てください。ANDゲートの入出力端子にすべてNOTを入れてみます。論理表を追っていくと，なんとORの論理とまったく同じになってしまいます。同様にORゲートの入出力にすべてNOTゲートを入れると，ANDの論理と同じになります。これを式で表すと，

$$(\overline{\overline{x}\wedge\overline{y}})=x\vee y$$

$$(\overline{\overline{x}\vee\overline{y}})=x\wedge y$$

となるのです。これはド・モルガンの法則と呼ばれるもので，これに従いXOR回路のORゲートの部分を置き換えてみたのが図6です。こうするとANDゲートとNOTゲートの2種類だけでXORを表現することができるのです。

しかし，これではICの数が減った分だけ，配線する箇所が増えてしまいました。結局手数として変わりなければ，わかりやすい回路のほうがミスが少なく製作することができます。では，元の図6(1)の回路のほうがよい回路なのでしょうか？ ということで，もうひと工夫してみます。図6(3)のように囲ってみると，同じパターンの回路が見えてきます。このパターンをどうにかいじることはできないか考えてみます。

この回路を図7のように変形していきますと，ORとNOTの組み合わせになっていて，これはNORゲート(図7(4))という比較的一般的な回路になるのです。これ自体がLS36というひとつのパッケージになっており，ANDとNOTをつなぐ部分の配線を省略することができます。このようにして工夫したXORは最終的に図8のようにまとめることができます。

ここまでの様子で想像できるかと思いますが，同じ機能をもつ論理回路の設計の仕方はひととおりでなく，いろいろなバリエーションが考えられるのです。論理回路を製作するときには，それぞれに応じて都合のよいものを選ぶことができます。では，ひょっとするとXOR全体がひとつのパッケージに入っているものがあるのではないか？ こう疑ってみた人はセンスのある人です。いまでは，デジタル回路の普及はめざましいものがあり，実際のところ，このハードウェア工作入門で扱う程度の基本的な回路は，ひとつのパッケージで実現できるICがすでに製品化されている，と考えてもかまわないほどです。

問題のXORゲートはLS86という型番のものです。回路図にも専用のゲート記号があります(図9)。これを使えば桁上がり付き1桁加算器の下位はLS86 1個，上位はLS08 1個で実現できます。

桁上がり付き2桁加算器

実際に計算器に使う場合，1桁どうしの加算器ではなんの発展性も期待できません。そこで，桁上がり付きで2桁以上の加算器を設計する必要があります。基本的には1桁加算器を並べるだけなのですが，2桁目以上は下の位からの繰り上がりも足し込む必要があります。1桁加算器の実際の製作に入る前に，その部分を設計しておきましょう。

まず，論理表を作ります。繰り上がりをキャリcとしておきましょう。和の部分は簡単で，xとyとをXORで足したあと，さらにcもXORで足せばよいだけです(表1)。

キャリの部分はx, y, cのうちどれか2つが1であれば，繰り上がりが生じることになります。そこで，xとy，xとc，yとcの3つの組み合わせについてANDを取ると，2つ以上が1の場合にかぎりそれら3種類のAND(表2の論理表でu，v，w)の中で1が残ります。もしx, y, cの2つ以上が1でなければ，これら3種類のANDはすべて0になるのです。そして，最終的にu, v, wのORを取ってやれば，それが次の上位へのキャリとなります。

以上のキャリ部分の説明を回路図に表す

図6 ゲートの置き換え

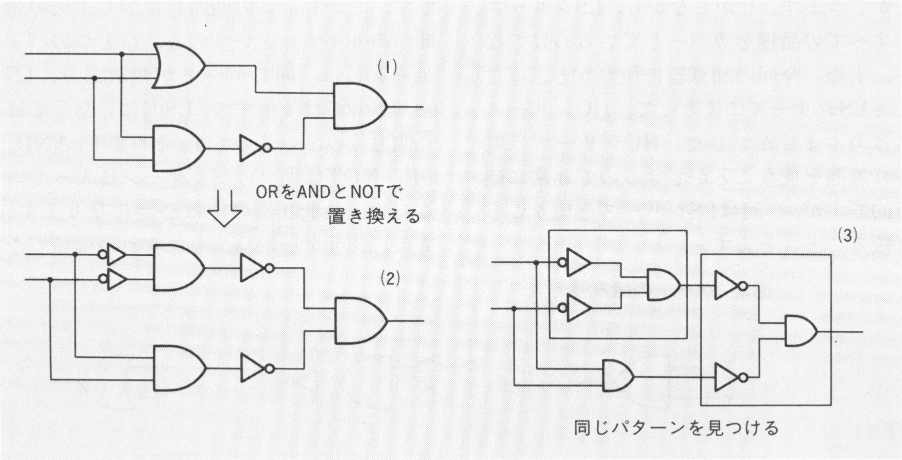


図7 回路の変形

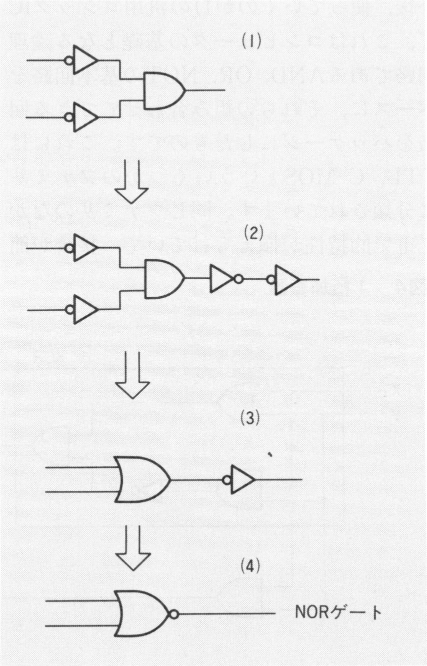


図8 XORの簡単な形

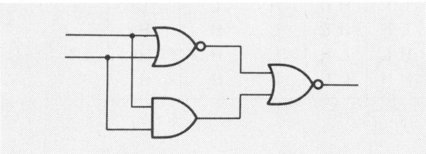
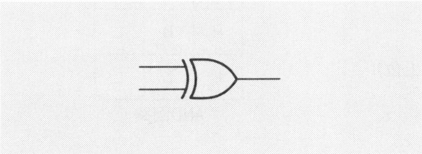


図9 XORゲート



と，図10のようになります。和の部分は先ほど登場したXORゲート２個，キャリ部分はANDが３個とORが２個の組み合わせになっています。２桁分すべての加算器を回路図にしたのが図11です。これを３桁以上の加算器にするには，図10（図11の点線部分）を付け足していけばよいだけです。

2桁全加算器LS183

さて，いま設計した回路をもう一度よく眺めると，どうも２桁目以上のキャリ部分（図10）だけゲート数が多いように見えます。この部分もっと簡単にできないかと普通は考えるのですが，残念ながらもうひと工夫したとしても，２個のORゲートの組み合わせを１個の３入力ORゲートに変更するだけしかできません。さらに実際には３入力ORゲートというICは，製品になっていないので４入力ORゲートを流用するしかありません。いずれにしても，必要なICの個数は，XORゲートが３個，ANDゲートが４個，ORゲートが２個ということで，それぞれ１パッケージですみます。しかしながら，この回路を実際に作ってみたところ，配線の数がかなり多く，１桁増やすごとの労力もばかになりません。

先ほど，この入門記事で製作する程度の回路はひとつのパッケージでできるものが多いと書きましたので，もしかすると，２桁加算器もすでにひとつのパッケージに収まっているものがあるかもしれない，と考えてみるのも悪くありません。実際のところ，桁上がり付き加算器はコンピュータ回路でも基本的なものなので，以下に挙げるようないくつかの品種が見られます。

LS180　1桁全加算器が１個
LS182　2桁全加算器が１個
LS183　4桁加算器が１個
LS282　1桁全加算器が２個１組
LS283　4桁全加算器が１個

このうちよく使われるのは，LS183とLS283です。今回はこれまでに設計してきた２桁加算器と同じ働きをするものとして，LS183を扱うことにします。図12はLS183の仕様を規格表から抜粋したものです。このICは14番ピンのVccと７番ピンのGNDとを別にすると，１～６番ピンまでの組と８～13番ピンまでの組とでまったく同じ回路

表1　繰り上がり付き加算の論理表

c	x	y	x＋y	z=(x＋y)＋c
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	0	1

表2　繰り上がりを調べる論理表

c	x	y	u=x∧y	v=x∧c	w=y∧c	c′=(u∨v)∨w
0	0	0	0	0	0	0
0	0	1	0	0	0	0
0	1	0	0	0	0	0
0	1	1	1	0	0	1
1	0	0	0	0	0	0
1	0	1	0	0	1	1
1	1	0	0	1	0	1
1	1	1	1	1	1	1

図10　キャリ付き１桁加算器

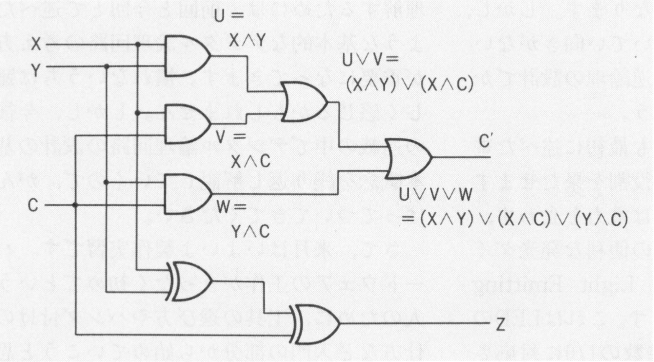


図11　2桁全加算器

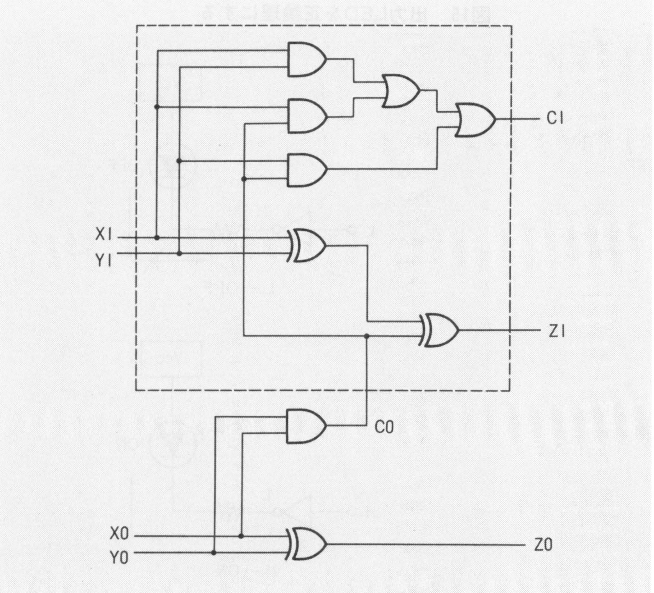
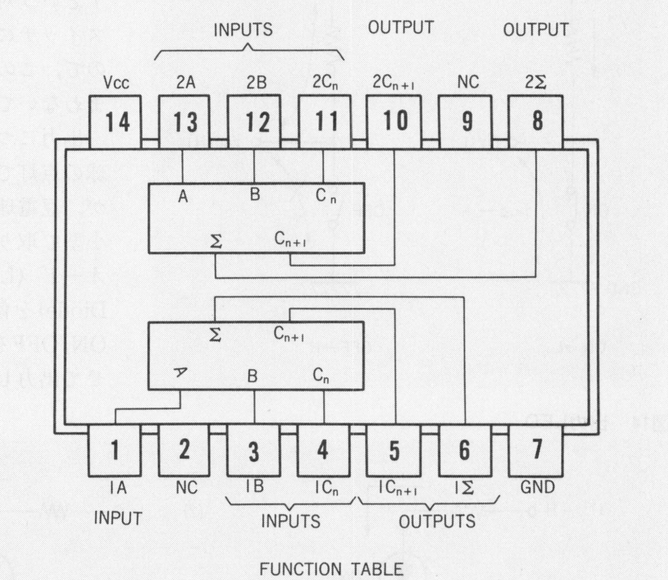


図12　LS183の規格表



FUNCTION TABLE
(EACH ADDER)

INPUTS			OUTPUTS	
C_n	B	A	Σ	C_{n+1}
L	L	L	L	L
L	L	H	H	L
L	H	L	H	L
L	H	H	L	H
H	L	L	H	L
H	L	H	L	H
H	H	L	L	H
H	H	H	H	H

完全に独立したFull Adderが２個入っている

が2組入っています。それぞれの回路は図10の回路と完全に対応しています。要するに，このICをそのまま使えば，ほとんど配線することなく目的の加算器を完成できるのです。

入出力インタフェイス

市販の加算器パッケージを使うことで，論理回路部分は簡単にできることがわかりましたが，実際に計算するデータを入出力する部分がないと加算器として働かせることはできません。そこで，次にデータの入出力を行う回路を設計することにします。

入力に関していえば，通常のコンピュータではキーボードとマウスが一般的です。しかし，この加算器では2桁の2進数を入力するだけなので，今月の最初のほうで説明したようなスイッチを使って入力する方式で十分だと思います。スイッチはIC用基板の上に直接取り付けられるタイプのものが便利です。それには，少し小型ですが，ディップ（DIP:Dual in Line）スイッチというデジタル回路用のスイッチが最適です。これは2～8個のスイッチが1列に並んだもので，2～8ビットデジタルデータの入力に使います。ここではディップスイッチのON/OFFを2進数の1/0に対応させて入力します。

スイッチを使ってデジタルデータを入力する方法として，図13の回路が非常によく使われています。ただし，この回路はスイッチのON/OFFと論理回路のH/Lとが入れ替わるため，論理が逆転してしまうので注意が必要です。図13(1)のようにスイッチがONのときは，入力端子がGNDとショートしてしまうので，入力端子にかかる電圧はLレベルになってしまいます。逆に図13(2)のようにスイッチがOFFのとき，入力端子は(抵抗を介して)Vccにつながりますので，入力端子にかかる電圧はHレベルになります。このようにON→0，OFF→1という対応になります。しかし，スイッチにはたいてい向きがないので，このまま逆論理の設計でかまわないでしょう。

出力についても最初に述べた電球の点灯で十分役割を果たせますが，豆電球よりは電力を食わず，小型で取り扱いの便利な発光ダイオード（LED：Light Emitting Diode)を使います。これはLEDのON/OFFを2進数の1/0に対応させて出力します。この場合，LSシリーズを使う場合は取り出せる出力電流の関係で，図14(1)，(2)のような接続はできません。基本的に使われる回路は図14(3)，(4)のようなものです。

これは，(1)，(2)のようにTTLICから流し出せる電流が0.4mAであるのに対し，(3)，(4)のように流し込める電流が8mAと余裕があるためです。しかしながら，この回路も論理が反転してしまい，0→ON，1→OFFの対応になってしまいます。今回の回路は，練習のために簡略化しているので，0→ON，1→OFFという対応だと割り切って考えればそのままにしておいてもよいのですが，どうしても1→ON，0→OFFにしたい場合は，図15のようにNOTゲートを1個はさむことで問題は解決します。

＊　＊　＊

以上で目的の桁上がり付き2桁加算器の回路が設計できました。全体の回路図は来月あらためて掲載しようと思います。いろいろと論理ゲートの組み合わせをいじくりましたが，結局は出来合いのパッケージを使うことに落ち着いてしまいました。しかしながら，コンピュータアーキテクチャを理解するためには，前回と今回とで述べたような基本的なデジタル論理回路の考え方が重要になってきます。慣れないうちは難しく感じるかもしれません。しかし，今後の連載の中でデジタル論理回路の設計の基本概念を繰り返し解説していくので，がんばってついてきてください。

さて，来月はいよいよ製作実習です。ハードウェアの工作がまったく初めてという人のために，工具の選び方やハンダ付けの仕方など入門の部分から始めていこうと思いますので，ぜひお楽しみに。

図13　入力スイッチ

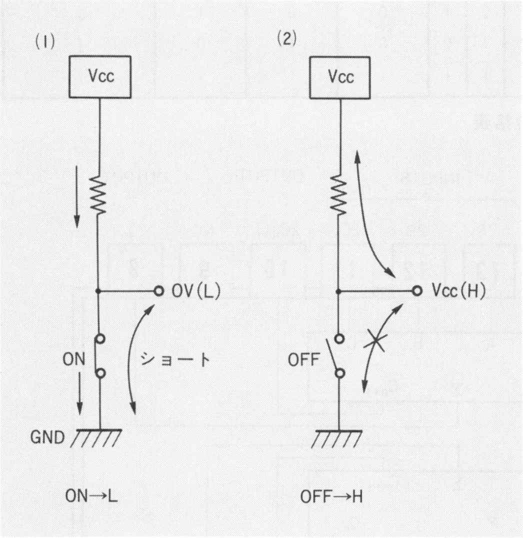


図14　出力LED

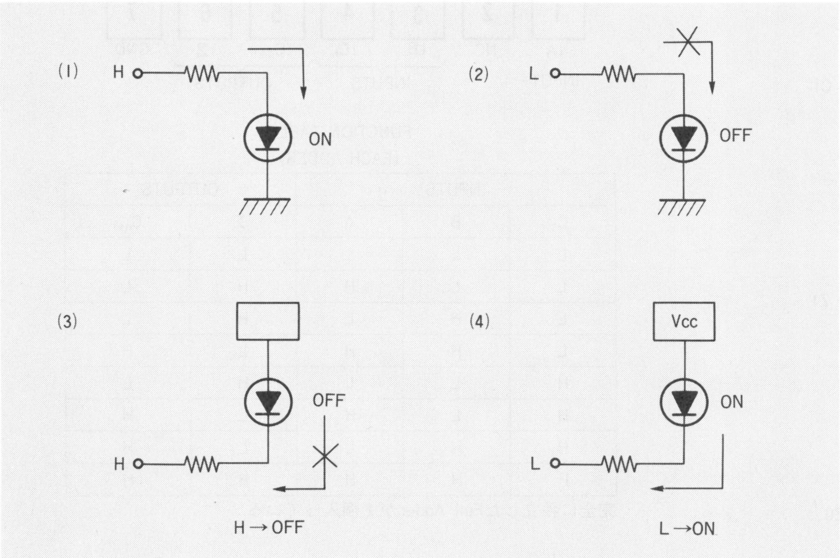
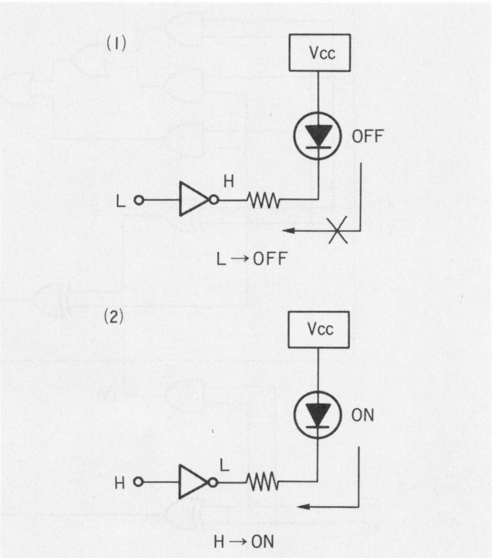


図15　出力LEDを正論理にする



CREATIVE COMPUTER MUSIC

Creative Computer Music入門(12)

偶成和音と借用和音

Taki Yasushi 瀧　康史

今月で一応基礎編は終わりです。いままで小難しい理論ばかりで疲れたでしょうが，音楽をやっていくうえでの基礎知識は十分身につけられたと思います。さあ，皆さんもどんどん自分自身で曲を作っていきましょう。

月日が流れるのは早いもので，私がこの連載を始めてから，すでに1年が過ぎようとしています。なーんて，この1年って結構長かったよー。だってほら，ネタのほうは意外とポコポコ出てくるもんだけど，サンプル曲の説明の苦しさ。「あとは楽譜を見てください」といって逃げちゃいけないんだけど，結局そうやってトンズラしちゃったりとかあったりして。うーん，もうちょっとボキャブラリーを増やす必要があるよな，私ってば……。

まあ，この1年間についての反省点は数々あれど，とりあえずなんとか1年お役目を果たせることができました。月並みな言葉ですけど，みんな，みんな，読者さまさまのおかげです（編集者さまのおかげでもある)。感謝してます。はい。今後もよろしくお願いします。

さて，いつまでも感慨に浸ってると，無駄話の好きな私は，いつまでもしゃべりかねないので，ここらで毎度のCD紹介をすることにします。

実は，今月はちょっと妙なものに手を出しました。最近よくある環境音楽の類です。どうもこれには2種類ほど分類があるようですね。

ひとつは完全にオリジナル曲で，なんだかよくわかんない曲がえんえんと流れています。私はこのテのことは詳しくないんですけど，なんでもサブリミナル効果だとかいう，人間の不可聴域である一定の音波を鳴らして，人の脳裏に直接訴えかけるような怪しい技術だそうで，人によってはかなり効くらしいです。そういえば，そんな感じかなーって雰囲気。こういうのって騙されてしまうほうが得ですよね。目が覚める飲料水とかと同じでね。

で，もう片方はリラックスしてしまうような穏やかな曲ばかり集めているようなものです（版権の関係でクラシックが非常に多い)。まあ，のんびりできる曲が詰まっていて（しかも芸術性うんぬんよりも，人間に聴きやすい曲が集めてある)，これはまあ，こんな感じかなーって雰囲気でした。

まあ，同じ「環境音楽」という題目の異なる2種のCDを2枚聴き終えたら，なんとなく「のほほん」って感じになってる気もしないこともないんですけどね。特にクラシックのほうは，聴きやすい曲のセレクションという向きもありましたし，興味があったらそこらへんでいっぱい売ってるのでいろいろ買ってみてもいいかもしれません。

さてと。前置きはここらで終わりにして，本題に入りましょうか。今回は，前回予告したとおり，偶成和音，借用和音についてお話ししたいと思います。

§ 偶成和音

偶成和音の名からもわかるとおり，この和音はもともとは偶発的に出来上がった和音です。古典的な和声法では，偶成和音は和音と名がついても，和音の仲間としては扱われていません。この理由はこれからお話しするなかから，徐々にその意味を感づいてもらったほうがいいでしょう。

ほんとは，これをお話しする前に4声体をお話しすべきなんだろうけど，はっきりいってこれは面倒ですから，ある程度簡単にお話ししたいと思います。4声体とは，簡単にいってしまえば，コードの進行をそれぞれの構成音ずつの進行と考えて，ソプラノ，アルト，テナー，バス進行の4つで考えるものです。とりあえず，いまはそうとだけ頭に入れておいてください。

まあ，それに，私の個人的見解からすると，偶成和音なんてあくまでも「偶然」できるんだから，意識しながら使わなくてもいいと思いますし。先月のように偶成和音がわりと使われてる曲を採譜するなら，そのとき，意識して「あ，これは偶成和音なのか」って改めて覚え直してもいいし，自分で曲を作るときに進行を見て「これは偶成和音を使えるな」と，ふと思い出して使ってみたりしてもいいです。とりあえずは，そのはしり的にそんなものがあるという意味でとってかまわないでしょう（もっとも，しっかり理解するには，自分でわざと偶成和音を使った曲を作ってみるのがいちばんですが……)。

保続による和音

余談がすぎたところで，本線復帰しましょう。まず，「保続音」という言葉ですが，この「保続」の意味を文字どおり取ると，「保ち続ける」ということになりますよね。これが，音という文字に修飾しているってことはすなわち，保続音とは「ある一定の音を保ち続ける」ということ，というのが妥当な解釈でしょう。

ところで通常，曲の進行の最中に，ある一定の音が鳴りっぱなしになったら，どうなるでしょうか？　はっきりいってヤですよね。耳障りですよね。耳障りにならないためにはどうすればいいか？

まず伸ばす音についてですが，いちばん耳障りにならない音があるとすれば，それは，曲の根音（すなわちスケールがCならC)，そしてあとひとつ考えられるのが，属音（5度の音，すなわちスケールがCならG）です。

また音の高さは，高めの音と低めの音とどちらがいいかというと，理由はややこしくなるので割愛しますが，どちらかというとこれは低い音，いちばん低いベースノートのほうがいいのです。

さて，保続されることがどういうことになるのかわかった時点で，ここで和声進行上で例をおいて考えてみましょう。

まず，図1，2を見てください。これらはどちらも同じコード進行ですが，一部のコードが図2では転回形に化けています。どちらもC調で書かれていて，図1が保続のない通常の進行，図2が保続のある進行です。図2中で音符に「ホ」と書いてあるのが保続音ですから，念のため。両者を聴き比べて（リスト1，2）みてください。このテのただコードが連続しているだけの例では雰囲気はなかなか伝わらないでしょ

うが，保続音を使う目的は，1度または5度の音を聴き手に意識させて，調性をわからせようとするものです。そんなふうに聞こえませんか？

さて保続音をまとめると，1度または5度のベースノートを動かすことなくそのまま保続させ，聴き手に調性をはっきりと理解させる，ということになります。当然，保続音のうえにわかりにくい進行をさせて，聴き手を困惑させるような使い方はバツです。

非和声和音

スケールC上でⅠ(C)からⅣ(F)への進行をするときのことをイメージしてください。この3和音の構成音は，C(maj)ではC・E・G，F(maj)ではC・F・Aです。

図1　保続のない通常の進行

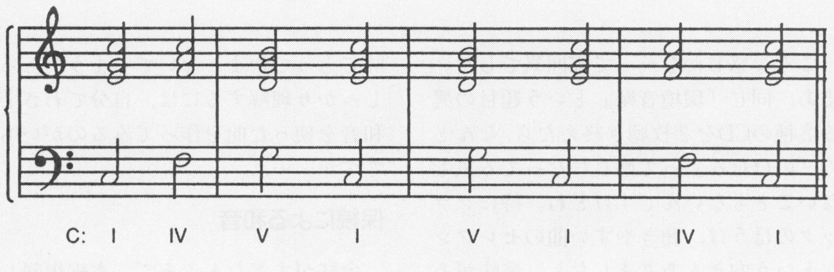
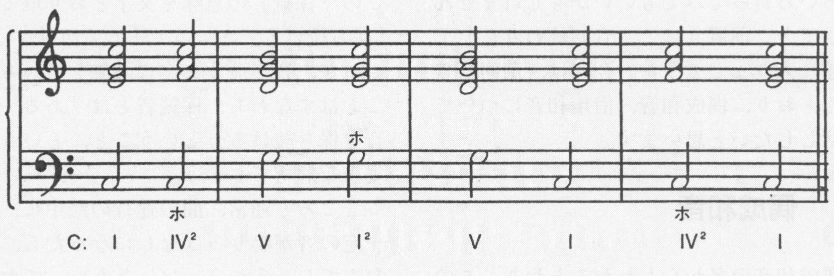



図2　保続のある進行

リスト1

```
(i)

.comment 図1保続のないもの

(m1,300) (aFM1,1)
(m2,300) (aFM8,2)

(t1) @1v15l2o4'eg<c''fa<c' 'dgb''eg<c' 'dgb''eg<c' 'fa<c''eg<c'
(t2) @1v15l2o3 c      f      g      c      g      c      f      c

(p)
```

リスト2

```
(i)

.comment 図2　保続のあるもの

(m1,300) (aFM1,1)
(m2,300) (aFM8,2)

(t1) @1v15l2o4'eg<c''fa<c' 'dgb''eg<c' 'dgb''eg<c' 'fa<c''eg<c'
(t2) @1v15l2o3 c      c      g      g      g      c      c      c

(p)
```

図3　コード上の単音レベルでの移行

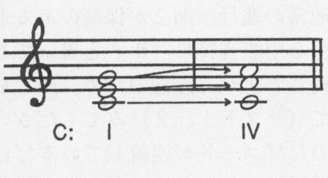
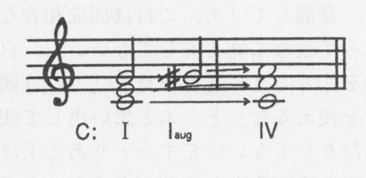


図4　経過和音を利用した移行

ここでこの進行を，それぞれの音のレベルで考えてみると，CはCへ，EはFへ，GはAに移動することがわかります（図3）。このうち，G→Aへの音の移行を，滑らかにつなぐためにG→G#→Aと経過的につないだらどうなるでしょう？

結論からいうと，図4のように進行は見かけ上，Ⅰ→Ⅰaug→Ⅳと進行します。Ⅰaugは非協和音のため，単独ではカデンツに含めて進行することはできません。しかし，単独ではよい響きとはいえないこのaugコードも，前後関係の結びつきによっては，単なる基本的なカデンツで作られた進行では絶対出せないような，独特の雰囲気を醸し出すことができるのです。

augを使った偶成和音は，Ⅰ→Ⅰaug→Ⅳのほかに，以下のようなものがあります。

1)　Ⅰ→Ⅰaug→Ⅱ　　（図5）
2)　Ⅰ→Ⅰaug→Ⅵ　　（図6）
3)　Ⅴ→Ⅴaug→Ⅰ　　（図7）

では，簡単にこれらを説明しましょう。まず進行上では，1)はさっきのⅠ→Ⅰaug→Ⅳの進行のⅣがⅡによって代理されただけです。音が経過的になっているところはⅠの5度の構成音Gが，Ⅰaugの構成音G#，そして最後のⅡの構成音のAと結ばれているところです。

そして2)の最後のⅥですが，これはトニックの代理です（サブドミナントⅣの代理とも解釈できる）。やはりこれも，G→G#→Aと経過的になっています。

そして3)のドミナントモーションも同じように，D→D#→Eと経過的に連なっていくのです。

ここに挙げた例は経過的なものだけですが，実は非和声音と同じ数だけ，すなわち経過和音刺繍和音，倚和音，掛留和音，先取和音，逸和音の6つがあります。また，実際にはaugコードだけではなく，9th，7th，さらにはdimコードまで使われます。結構奥が深いんですよね。そんなわけでここで全部覚えるのもなんですから，ここではそんなものがあるということだけ覚えていてください。参考にツェルニー30番練習曲－26から，経過的半音階進行の例を載せておきます（楽譜1）。

§ 付加和音

1991年11月号で話したきり，一度も使い方に出てこない和音があります。dim，augは今回出てました。しかし，一向に出る気

図5　I→Iaug→II

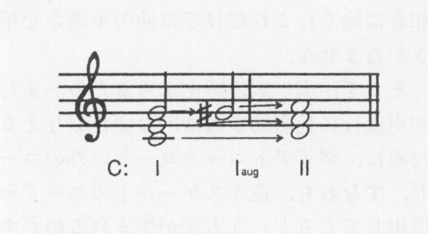



図6　I→Iaug→VI

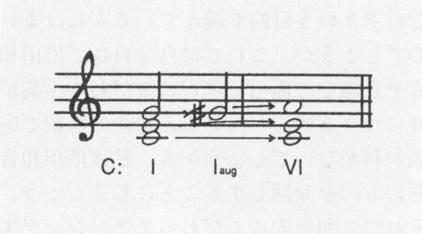
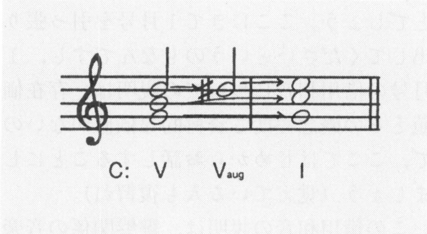
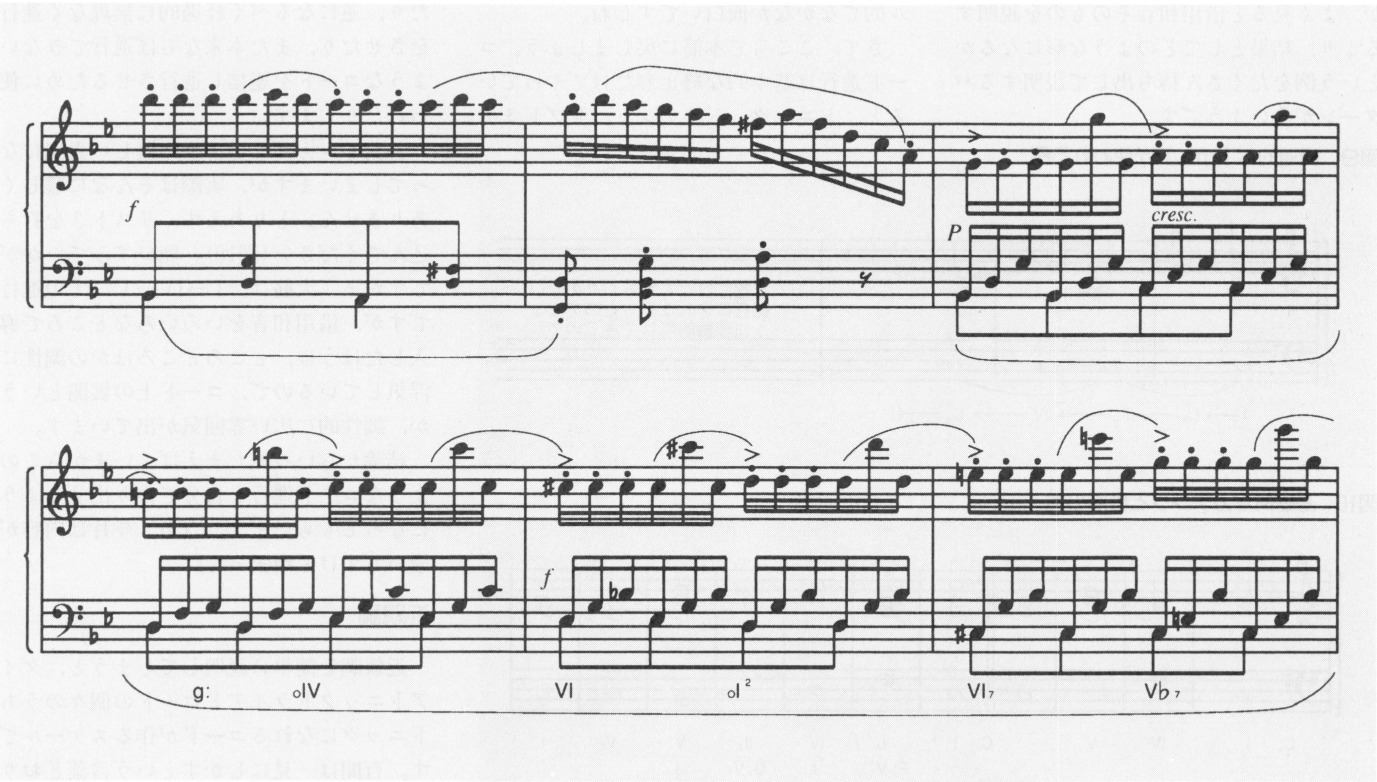



図7　V→Vaug→I



楽譜1

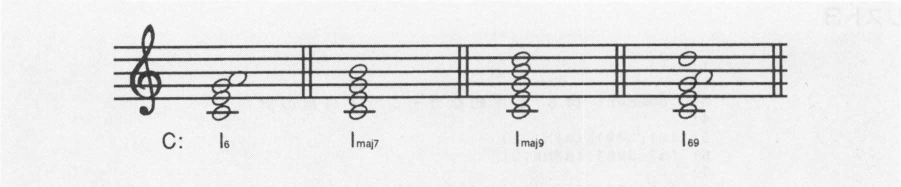


配も見せないI6，Imaj7，Imaj9，そして通称HコードといわれているI69などは，どうしちゃったんでしょう。

実はコードの種類については，いままでにやってきた一般的な終止形によるカデンツァ進行，偶成和音を使った特別な進行，そしてこのあとの借用和音で，進行のパターンも含めてすべて今回でひととおり説明しきってしまうのです（すなわち，アナリーゼはこれだけの知識でひととおりできてしまう)。でも，このあと説明する借用和音にも，新しいコードは出てきません。そうなると，例の4つはいったいいつ使われるのかと思うでしょうし，市販されている楽譜を読んでいて，このようなコードに出合ったらどのように解釈したらいいのか，という疑問もわいてきますよね。

さて，ここで図8を見てください。これらのコードの根底にあるトライアドコード（4音以上で構成されるコードは通常トライアドコードがもとになっている）を調べてみると（そんなのばっかり集めたのですが)，すべてIになるのがわかってもらえると思います。

図8　付加和音の例（トニックとして代用されやすい順）



すなわち，これらの6度の音，7度の音，9度の音は，緊張感を出すために単にIに付加されただけだと考えられるということです。C調での6度のA，7度のB，そして9度のDは，非和声音として扱う必要性はないのです。ただし，当然これらの和音はすべてトニックとして扱っていいとはいえ，やはりただのIとは違うので，それなりの対処が必要です。たとえば，I9では曲を終わらせないとか，念のためIに進行するとか，そういうことを行う必要性はあるでしょう。

ちなみに，これらの付加和音は時代的にかなり新しいコードですから，クラシックでは使われているものが極めて少ないですし，また使われていたとしても，トニックとしてではなく，経過的に連続して，I→Imaj7→I7→VI→Iaug→I（図9）という具合に（ずいぶん回りくどく書きましたが)，決してトニックとしては使われませんでした。

§借用和音

偶成和音とは違って，借用和音は完全に意図的に盛り込まれたコードです。

1月号でこれについてお話ししたときは，終止形を話した直後でしたので，よく意味

が取れずにそのままにしている人もいることでしょう。ここにきて1月号を引っ張り出してくださいというのもなんですし，1月号の借用和音についての説明は，存在価値とその概念程度で資料的な価値がないので，ここではじめからお話しすることにしましょう（覚えている人も復習ね）。

この借用和音の説明は，鍵盤関係の音楽雑誌で毎年のように行われているようですが，よく見ると借用和音そのものを説明するより，結果としてどのような形になるかという例をたくさん持ち出して説明するパターンが多いようです。

教える側でも楽だし，すぐに実用になるためそういう目的で読んでいる人にはよいのでしょうが，これでは自分自身で借用和音を意識して面白い進行にこだわる作業が身につきませんし，いろんな曲の解釈で応用が利かないでしょうから，私は借用和音そのものをお話しすることにしましょう。それに，和声的にこだわってアレンジをするという作業は，できるようになるとパズル的でなかなか面白いですしね。

さて，ここらで本筋に戻しましょう。コード進行は基本的な終止形だけでやっていると，トニック，ドミナント，サブドミナントの3つだけになってしまいます（代理和音は除く）。これだけでは曲の単調さを招きかねません。

そこで，広い調性感を求めるため，また，和声進行に色彩的で機能的な変化を与えるために，ダイアトニックコード以外のコード，すなわち，違うスケール上のコードを借用してくるという方法が生まれるのです。このことによって，曲にアクセントをつけたり，逆になるべく経過的に無理なく進行をさせたり，また本来ならば進行できないようなコードを連結し進行させるために使われたりします。

言葉でいえばこのように難しい表現になってしまいますが，実際はそんなに難しくありません。とりあえず，リスト3を打ち込んでください(図10)。聴いてみていかがですか？　大筋は，I→IV→V→Iの進行ですが，借用和音をいろいろなところで導入したほうは，ところどころほかの調性に浮気しているので，コード上の展開というか，調性的に広い雰囲気が出ています。

結論からいってしまえば，いまからこのようなコード進行を自分で作り出せるようになってもらいます。さあ，今月は内容が濃いですけど頑張って!

図9　経過的にⅠmaj7が使われる例

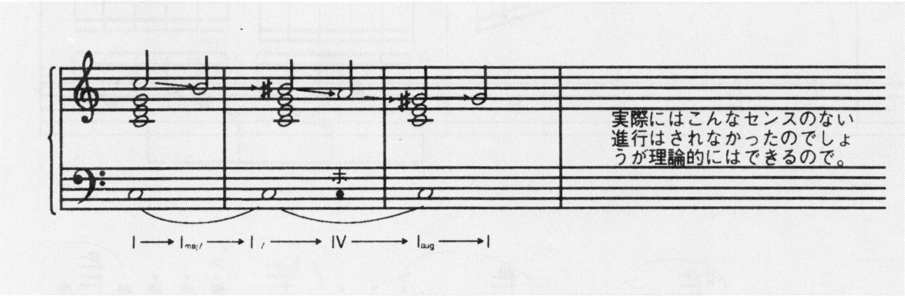
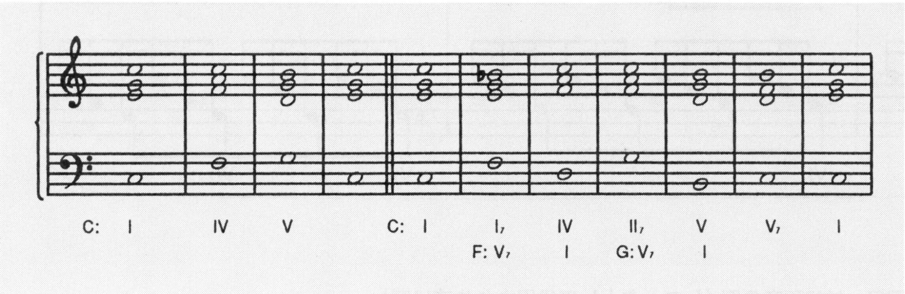


図10　基礎的なカデンツと借用の例



リスト3

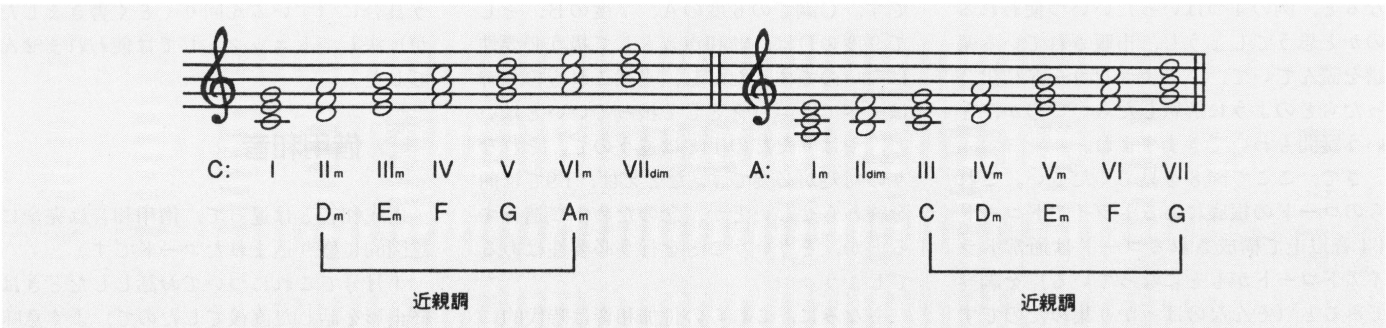
```
(i)

.comment 図3 基本的なカデンツと借用の例

(m1,300) (aFM1,1)
(m2,300) (aFM8,2)

(t1) @1v15l2o4'eg<c''fa<c''dgb''eg<c'
(t2) @1v15l2o3 c     f     g   c

(t1) r 'eg<c''egb''fa<c''fa<c''dgb''dfb''eg<c'
(t2) r  c     c    f     d        g   >b<   c

(p)
```

近親調

近親調を簡単に説明してしまうと，ダイアトニックトライアドコードの個々のうち，トニックになれるコードが作るスケールです。百聞は一見にしかずという言葉どおり，話に聞いてもわかりにくいでしょうから，図11を見てください。

わかったでしょう（と決めつけている）。このなかで，長調のVII，短調のIIは，減3和音（すなわちdimコード）となるので，トニックにならず，除外します。すると，C(maj)スケール上ではDm・Em・F・G・Am，AmスケールではC・Dm・Em・F・Gが近親調となります（注：近親調の求め方は，このようにダイアトニックコードから導き出すようなものとは違って，調号の数から

図11　近親調



求めるものもあります)。

原調から見て，その調と調号(#，♭)の数が同じか，あるいはひとつ違いのものが近親調です。すなわち，調がCmなら♭が3つですから，当然同じく3つのE♭(maj)は近親調，そして，♭が2つのB(maj)，Gmも，4つのA♭(maj)も，Fmも近親調ということになります。

どちらのほうが求めやすいかはその人しだいになりますが，調号がどこにつくかまだピンとこない人にはこの方法がよいかもしれません。

借用和音になりうる和音

借用和音で最も代表的なものは次の2つです。

1)　近親調から借用された副Vの和音
2)　同種短調から借用された準固有和音

副V，同種短調，準固有和音などといきなりいってもわからないでしょうから，順に説明していきましょう。

まず副V，これは近親調のVです。Vだけではなく，V7，V9，そしてややこしくなるのであまり述べたくないのですが，V7，V9の根音省略系というものがあります(こういうのをV諸和音という)。

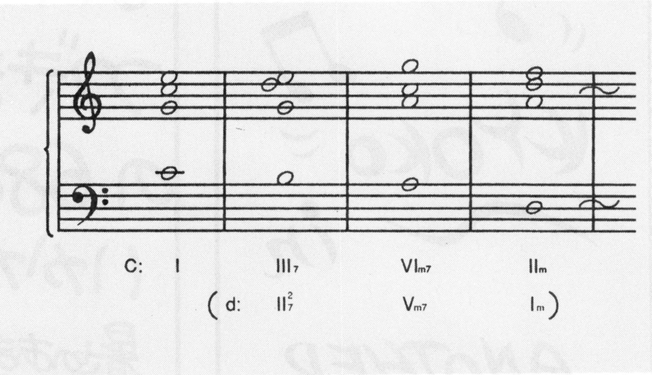
同種短調とは，C(maj)スケールに対するCm，G(maj)スケールに対するGmです。

そして，準固有和音とは，いま述べた同種短調の固有和音(ダイアトニックコード)のことです。

とりあえず，上の2つのうち，「近親調から借用された副Vの和音」だけ，今回はやることにしましょう。「同種短調から借用された準固有和音」は，いずれサンプル曲かなにかで出てきたときに説明することにします。

さて，借用和音の奥義は，実はただのドミナントモーションです。ドミナントモーションというのは，もう皆さんわかっていると思いますが，ドミナントはトニックに進まなくてはならない！　という法則で，この性質を利用して，近親調の和音を借りてこようというのです。

たとえば，さっきの図10の6小節目のC7は，スケールC上から見たらI7にすぎなく単なる付加和音なのですが，C(maj)スケールの近親調のF(maj)スケールでは，これは，V7でドミナント7thです。したがって，ドミナントモーションが起こり，V7はトニックIに進みたがります。よって，F(maj)スケールのトニック，F(maj)に当然のように進みます。この動きを原調C(maj)から見たとき，I→I7→IVという進行になるのです。これが借用和音の基礎です。注目すべきところは，C(maj)にはない音がF(maj)には含まれていることです，この場合はB♭がそうで，このように原調にはないけれど，借用された近親調にはある音を特徴音といいます。

図12　D2の使用法

さて，いまの例の応用として，原調(ここではC)のV(7)に滑らかに進むには，前にどんな借用和音を置くといいか考えてみましょう。原調(C(maj))から数えてV調(G(maj))すなわちGのドミナント7thはD7ですよね。このD7は原調から数えるとII7。この進行は原調から見れば，

(a)　II(m)(7)→V(7)→I

なんと，1月号でお話しした終止形のひとつになってしまいます。あまりにも日常的な進行なので，終止形の一部に数えられているのです。でも，その基本は借用和音からきているのですね。

また，このII(m)(7)はD7調のD7なので，ドッペルドミナントとか，ダブルドミナントとか，セカンダリードミナントとかいわれています。記号はDが2つ連なった形や，D2と書いたりするのですが，印刷の都合上(タイプの都合上)今後これを引用するときは，この講座ではD2(こちらのほうが原調のIIだとこじつけができて覚えやすいでしょう)とすることにしましょう。

D2をさらに使う

ここで，D2は単なる借用ではなく，終止形のひとつの形にすぎないんだと考え，このD2を利用した効果的な進行の例を考えてみましょう。

先ほどいったとおり，借用和音のもっとも一般的なものは副Vを使うものでした。この考えをさらに発展させ，(a)の進行が一般的なものだと意識するなら，このII7も借用に使うことが可能だといえます。

ここで図12を見てください。IからIIに進行するとき，(単純に進めますが)借用和音を使って進行するとすると，

C：　I→III7→VI(m)(7)→IIm

というように進行できます。このままでは理解しにくいでしょうから，

C：　I→III7→VI(m)(7)→IIm
(d：　II27→V(m)(7)→Im)

というふうに表してみます。カッコの中のdとは，Dmスケールからコードが借用されていることを表します。ちなみに，dが小文字なのはスケールがマイナーだからです。これがメジャースケールなら大文字で書きます。

この例を見て，そして聴いてみてわかるとおり，実際のところD2はさして意識もせず，Dの前にあるものとして，使われていることがわかります。

借用和音は副Vだけではなく，Vの前に進行する何かを置くことも可能なのです。ここでの例は借用された近親調のD2(副II)でしたが，実際にはそのほかにS(サブドミナント：副IV)なども使われます。

§まとめ

編集部から，基礎理論は今月までにして，来月からは応用編をやってくれという依頼があったので，今回はそのしわよせで，理論中心のお話になってしまいました。あんまり今月みたいな調子は，好きじゃないんですけどね。まあ，先に進むためにはしかたがないでしょう。

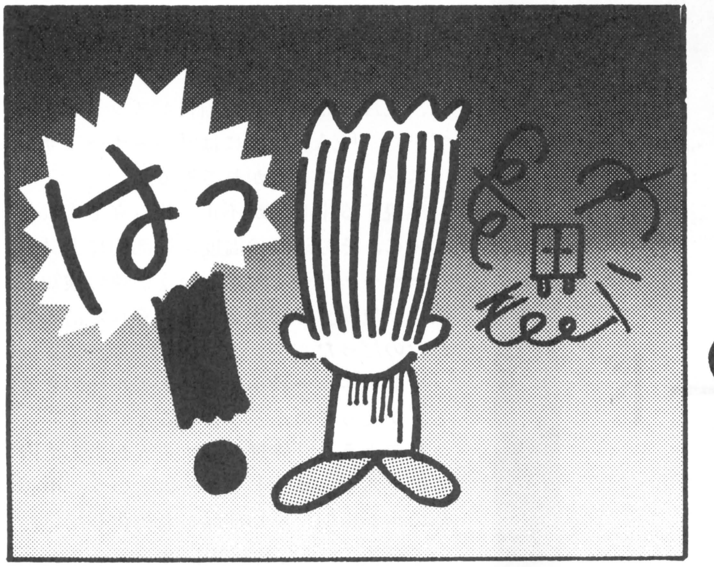
コードブックを読んで途中で挫折してしまったような初心者にありがちな，「トニック，ドミナント，サブドミナントの機能はわかったけど，それ以外はどうやって使うの？」みたいな素朴な疑問も今回で解けたことと思います。

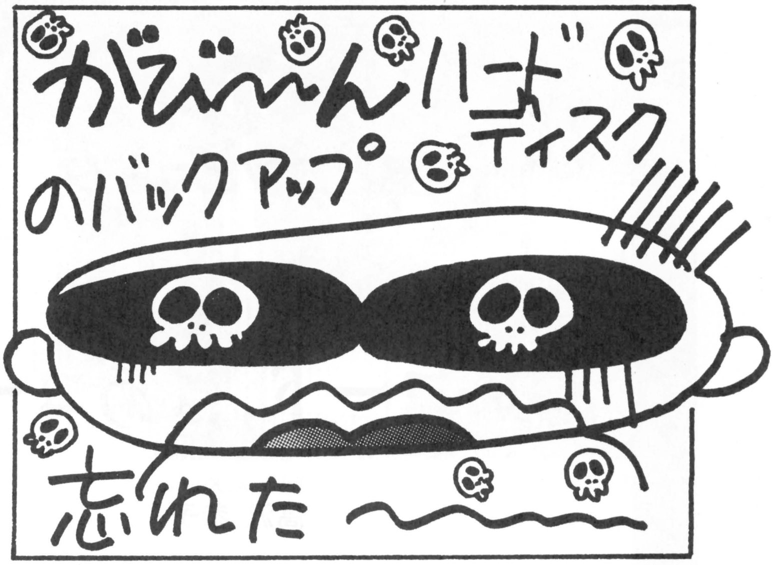
さて，来月ですが，ひょっとしたらお休みするかもしれません。応用編ではできるだけ，すぐに使える応用テクニックをやってみたいのでネタの整理をやらなきゃならないんです。でも，基礎知識総集編みたいな感じで「資料」として，1年分の原稿をまとめるかもしれません(保証はできませんけど)。

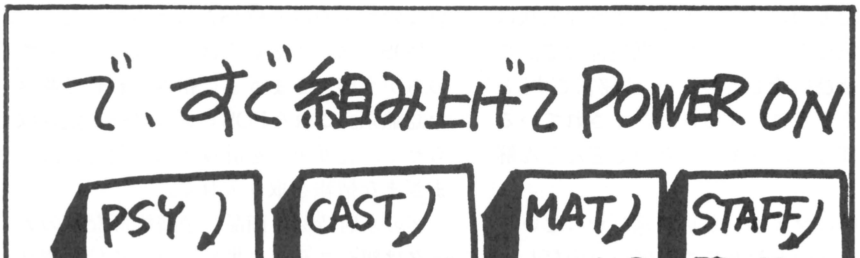
ま，そんなわけで，今月号で基礎編は終わりと。

ではまた，来月か再来月にでも会いましょう。

KYOKO in ANOTHER CG WORLD
ごきぶりではありません
まいにち残暑がつづきますが… みなさんの68の具合はいかがでせうか…
暑さのあまり暴走〜〜っつうことはないみたいだけど
68
えーわたくしめは、つい最近お引越しをいたしました…
エーイなんでこんなにものがあるんだろー
FDの山
よろしくおねがいしまーす







今回のCGデータ

総物体数　382

うちメタボール数　60

光源　22

1280×1024ピクセル

1670万色フルカラーを

４×５ポジで出力

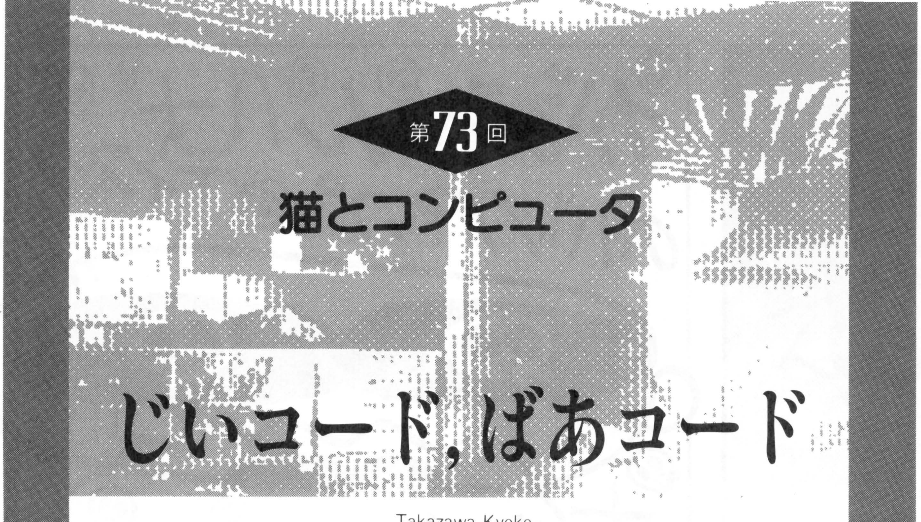
使用ソフトは

C-TRACE, サイクロン

マッピングデータ作成に

Z'sSTAFF PRO-68K

今回はお引越しと
透明体が多くて
レンダリング時間が
かかったのとで、
大巾にメ切に遅れて
しまった… 編集者の
Aさん ごめんなさ～い
(っていつも言ってるよーな気もする)



第73回

猫とコンピュータ

じいコード，ばあコード

Takazawa Kyoko
高沢 恭子

最近何かと世間を騒がせている「バーコード」と「Gコード」。なんにでも興味を示すキョウコさんらしく，「Gコード」を入力する「ビデオプラス」を購入していろいろと実験しているようですが……。

灼熱の部屋

「ここもあけたんですか？」
「ハイ……」
円筒のカバーがはずされた。
「これもですか？」
「ハイ……」
金網のカプセルが取られた。
「もっと中までですか？」
「ハイ，はずせるところまで」
小さな円盤もはずされた。
「あー，ここが逆です。あけてみてよかったあ。ふつうはここまでは調べないんですけどネ」
こうして7月の午後のリビングでファンヒーターは焚かれた。試運転だ。
「15分くらい燃やしてみないとわかりませんから……」
SANYOの修理マン氏は汗をふきふき，そういいながら麦茶をひと口飲んだ。
夏場は暖房器の修理は受けないようにしているということだったのに，運よくきてもらえた。そして，やっぱり原因はアレだったのだ。
炎をみつめて何分もじーっとガマンの観察をしながら，修理マン氏が遠慮がちに聞いた。
「こういうこと，好きなんですか？」
“こういうこと”とは機械をバラしてみること。故障の最終的な原因は分解掃除のあとの私の組み立てミスだったのだ。
「あんまりいろいろなエラー表示が出るものですから，どうしてもあけてみたくなって，誰だってそうじゃありませんか？」
「あまり女の人はねえ，中まであけてはみないと思いますけど……」
「はい……。そうですよね」
春が近くなるころから，ファンヒーターにエラー表示が多くなった。「E2」やら，「E7」やら，そのつどエラーの内容を示す番号がちがうし，「換気」のアラームがひんぱんに鳴るなど症状もとらえにくい。
ボディカバーをはずし，中のマイコン装置をみたものの，目でわかるような異状はない。気になるのは中心の燃焼部分で，この中をぜひ掃除してみたい。“秘密です”とでもいうように，すっぽりかくされている円筒の部分を，メモもしないでどんどん解体した。
この中はマイコンと無関係だからだいじょうぶ。元のとおりにすればいいのだし，プラモデルよりやさしいと，汚れを取ったあと再び組み立てた。
このとき1カ所だけ部品の向きに不安があったが，ひとまずそれらしい形に戻して試運転。結果，いままでの不安定なエラーはすべて解消した。ところがそれも束の間，点火して10分くらいすると，こんどはマニュアルにない「EA」というエラー表示が出て，消火してしまうようになった。
どこかマイコンの配線を損傷してしまったのだろうか。でもあれだけたくさんあったエラーがひとつにまとまった。ただし，前より悪い状態になったのかもしれない。
「EAというエラーはマニュアルにはないんですけど」と私がいうと，
「メーカーのマニュアルにはあるんです」と，修理マン氏がそれをみせてくれた。
「虎の巻」だというわりに簡単な印刷物には，「EA」「EP」などの項目がズラリと並んでいた。「EA」の欄には「バーナー・サーミスタ」とだけ書かれてある。
燃焼する中心部が不正に組み立てられたために，その近くの温度が局部的に上昇し，バーナー・サーミスタという燃焼の検知装置が「危険」の判定をした。それで消火されたのだった。
「全部のエラー表示をマニュアルにのせると，かえってめんどうになるんですよ」
私のように「修理」をする人がふえて困るのだろう。ロッドフレームという，中心部のネジに似た感知部品も老朽化していたので交換，しめて4,365円。オフシーズン手当は請求されていなかったようだ。

進化のあと

7月18日付の朝日新聞で，ファーストフードの廃棄率の話を読んだ。ハンバーガーなどで知られるファーストフードの各チェーン店では，調理されてからほんの短い時間を，「賞味時間」と定めて，その時間を超えて売れ残った商品は捨ててしまうそうだ。マクドナルドでは10分間が制限だった。
それでも1000個こしらえて5,6個捨てるくらい，つまり1％に満たないのだそうだ。そして，これを最小限に抑える役割をしているのが，やはりPOSシステムだった。
POSシステムといえばバーコード。太さのちがう棒をさまざまに組み合わせて並べた認識票。これをPOSレジスターに読み取らせて，売り上げを計算すると同時に，さまざまな情報を取り入れる。
客の来店時刻，商品，消費総額などのデータは親システムに集められ，日付や曜日，時間帯による商品の売れ筋などを分析して経営戦略に役立てる。
バーコードはPOSシステムと同時に，1982年ごろから急速に普及して，いまは400万以上の数があるそうだ。
このごろの子供の遊びはバーコードを使うと聞いて，これだけ氾濫しているものを遊びにする子供のウイットはすごいなと思ったら，なあーんだ，これもおとなが仕掛けたものだった。
オモチャのメーカーが，バーコードを数値に置き換えて処理をする，専用のゲーム機を発売したのだ。
カードに貼りつけたバーコードをゲーム機に読み取らせると，「生命力」「攻撃力」「守備力」などにして表示する。2枚のバーコードでその数値を戦わせて，「生命力」が先にゼロになったほうが負けだ。

ほとんどが男の子であるマニアたちは，バーコードを何百枚もあつめ，名人級の子はみただけでおよその強さがわかるそうだ。全国大会もあるという大流行ぶりだが，試合の特徴としては機械を介したゲームらしく，クールで，白熱することはないという。やっぱり子供たちがつくりだした遊びとは，このへんがちがう。

雑誌も薬もスナックも，何からなにまでバーコード。商品が戸籍をもっているようなものだ。スーパーでバーコードがないのはモグリか，日替わりの臨時商品だ。バーコードのない商品を買おうとすると，とたんに手動の入力になって，数秒間よけいに待たされることになる。

バーコードがスゴイのか，POSが偉いのか。一瞬の光を当てるだけの入力には，はじめは魔法をみたように驚いたはずなのに，みなれたというだけですぐに機械の優位に立ったと思う私たち。そして疑念ももたず，どんどんラクになる。

努力しないで何かを得てはいけないのだと，なんとなく子供のころから呪文をかけられたせいか，イソップやたくさんの寓話に洗脳されてきたためか，あまりに無為のまま成果を得てしまうと，いつか怠けたことの処罰が待っていそうな気がする。

便利なものがあらわれるたびに，人間は引き換えにひとつずつ能力を捨てていく。そういうことを「進化」と呼ぶのだといった人がいた。

捨てられたこととは……。

ナイフで鉛筆を削れること？　手で洗濯できること？　元気に長い時間歩けること？　それから，筆算ができること？　漢字が正しく書けること？　なんだか，私もだいぶ「進化」してしまった。

カギは時計だ

最近また私たちを進化させる新しいものがあらわれたので，いくつか購入して身近な人にプレゼントしてみた。誰が進化して，誰が現状にとどまるか。

それは，テレビ番組の録画予約をするための専用リモコン「ビデオプラス」という商品だ。そして，まっさきに活用をはじめたのが狛江のアニキだった。

ところが，使用数日でかけてきた電話では，10チャンネル（テレビ朝日）に予約の設定をしたものが，どうしても1チャンネル(NHK総合)に対応してしまうと，不可解な症状を報告してきた。

海外の製品なので，購入元から取次店に送り調べてもらったが，どうも原因がわからないという。修理の担当者とアニキとは，あれこれ電話で話し合った。そして思わず声をあげたのはアニキだった。マニュアルの欄外の小さな文字に気づいたのだ。

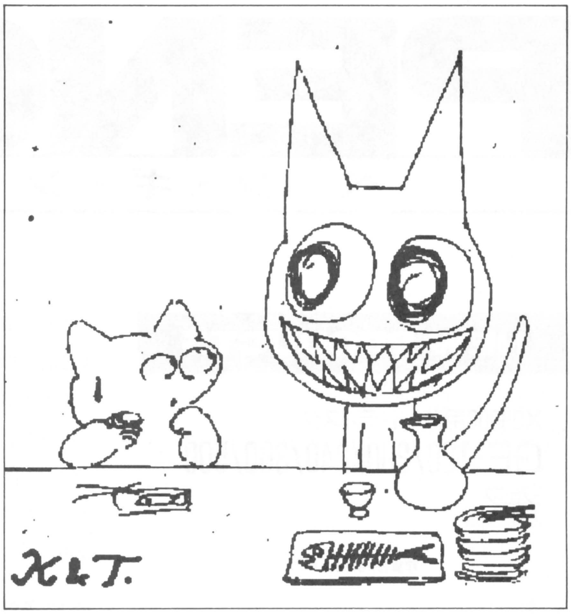


illustration : Kyoko Takazawa

予約機にはいくつか初期設定が必要だが，所有ビデオのメーカーを，決められた番号で入力しなくてはならない。アニキはビクターの番号とされている「01」を入力したのだが，欄外の注意事項にはビクターでもアニキの使っている「HR-S8000」という機種は「04」にするように記されていたのだ。

無事トラブル解決。進化のためには順応の努力も必要なので，いちばんのりのアニキはそれなりに偉い。ほかの数名はまださわろうともしないのに。

「ビデオプラス」は「Gコード」（ジェムスターコード）という番組ごとに定められた8ケタ以下の数字を入力するだけで，一度に8番組までの予約ができる。「Gコード」はいまのところ朝日新聞とテレビガイドに掲載されているだけだ。

ビデオの予約には，たいていの人が手順が複雑すぎるという意見をもっているらしい。この機械も，録画予約にことごとく失敗していたというエピソードをもつ，ヘンリー・ユーエンという香港生まれの数学者によって発明されたのだそうだ。

ことごとく失敗というあたりはわが家もレベルが一致。「ビデオプラス」は救世主といってもいい。ただ，なんとかもっと扱いやすくとは望んでいたものの，ここまでシンプルにされてしまうと，ありがたい半面，なさけないような気分にもなる。

各番組の末尾にゴシック体で記された数字。いちばん小さいものは2ケタもある。これを入力して録画ボタン（毎週，1回など）を押すと，放送日と録画開始時刻が表示される。どれを入力しても，確認すると番組表どおりの放送日時になる。1枚の新聞だけでも何百もあるGコード。このマジックはなんなのだ。

パソコン通信のボードでも，みんながかわるがわるいろいろな推論を立てていた。コードの中から何か規則性をみつけようとしたり，2進法をあてはめてみたり。

わが家でも，あれこれトライしてみた。でもナゾは深まるばかり。

夫が出張先の関西から持ち帰った新聞を広げて，

「みてごらん，各地にそれぞれテレビ局があって，すべてにGコードがつけられているんだよ」

ほんとうに膨大な数だ。いくつあろうと公式はひとつなのか。弁護士の資格ももっているという数学者は，いまごろさぞ満足に思っているだろう。

夫が，なにげなく前日の新聞に掲載されているGコードを入力してみた。予約の結果を確認してみると「Err」が表示された。その前日のコードは放送日時があらわれるが不正確。その前日も同じ。

そこで，こんどは内蔵の時計をある日時に戻して設定，その日からは未来となる放送日のGコードを入力してみた。なんと正しく予約の放送時刻が表示された。つまり時計を基準にして，将来だけについて予約のプログラムをするのだ。

ただし，無制限の未来というのはダメなようで，ずっと過去に時計を戻して実験をしてみたら，1カ月先くらいまでが限度だった。

遊んでいるうちにわかったのはこれだけ。Gコードの謎にはみんながファイトを燃やしているらしく，ジェムスター・ジャパン社にも，コードのつくり方を教えてほしいという電話がよくあるそうだ。

さて，やはりもう一段の進化をとげてしまいそうな私は，きょうもふしぎなGコードとにらめっこ。

「じいコード　に「ばあコード」。秘密のワザをもった仙人なのだろうか。

PENGUIN INFORMATION CORNER

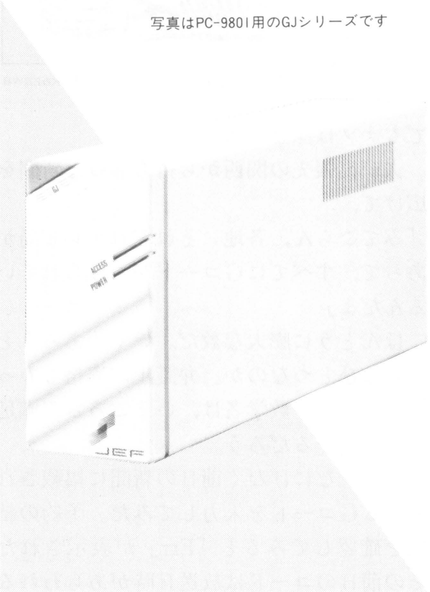
ペ・ン・ギ・ン・情・報・コ・ー・ナ・ー

NEW PRODUCTS

X68000用ハードディスク

GF-120/200/240/300/500

ジェフ



写真はPC-9801用のGJシリーズです

ジェフは7月下旬よりX68000のハードディスクを発売した。

本機は，WORKSseriseとして新しく発売されるもので，120Mバイト(GF-120)から500Mバイト(GF-500)まで5タイプ用意されている。

すべてのタイプに信頼性を高めるオートシッピング機能，バッドエリアを自動交換するインテリジェント機能を装備。外形寸法は60mm(幅)×120mm(高さ)×295(奥行)となっている。

キャッシュバッファとして「GF-120/200」は64Kバイト，「GF-300/500」は128Kバイト，「GF-240」は256Kバイト搭載している。

インタフェイスにはSCSIを使用。IDスイッチの変更により7台まで接続可能である。また，接続ケーブル（フルピッチ50ピン）とターミネータが同梱されている（インタフェイスボードは別売）。

価格は「GF-120」が108,000円，「GF-200」が138,000円，「GF-240」が148,000円，「GF-300」が318,000円，「GF-500」が418,000円（すべて税別）となっている。

〈問い合わせ先〉

㈱ジェフサポートセンター ☎06(336)5901

電子システム手帳ICカード

PA-3C45S/48S

ディ・メーレ/ログ

シャープ電子システム手帳用ICカード2種が発売される。

PA-3C45S

●ステラ薫子のミッドポイント占星学カード「PA-3C45S」（8,4行表示専用カード）

生年月日，出生時刻，出生場所を入れるだけで，固有のホロスコープ(天体配置図)をサーチし，本人の運気を割り出してパーソナルな占いを行うことができる。

また，利用者本人が過去の運気を質疑応答方式で入力していくことにより，占う人の思想や行動に影響を与える星（支配星）を見つけ，より精度の高い占いが可能となった。

人生総運，仕事運，金運，愛情運（男），愛情運(女)，相性診断の6種類の占いができ回答文章数1,120，運気の傾向調査の質問文章448を収録。

価格は9,500円（税別）となっている。

〈問い合わせ先〉

㈲ディ・メーレ ☎03(3468)7253

PA-3C48S

●イベントクイズカード「PA-3C48S」（8,4桁表示用カード）

テレビでお馴染みの篠沢教授とそのスタッフが考えた，問題数約3,000問のクイズゲーム。プレイヤーはクイズに答え，正解数だけビルを登っていくという設定となっている。そして，登場する3つのビルをクリアすると超難解な「スペシャルクイズ」50問に挑戦することができる。

また，ひとりで遊べる「チャレンジモード」と，2人でできる「対戦モード」の2つのモードを用意。クイズの合い間には，神経衰弱やスロットマシン，撃沈ゲーム，競馬ゲームなど9種類のイベントゲームが遊べるようになっている。

途中で電源を切っても再開できるコンティニュー機能や，カードを抜いても再開できるパスワード機能を搭載。ゲーム中にルールを確認できるヘルプ機能も装備している。

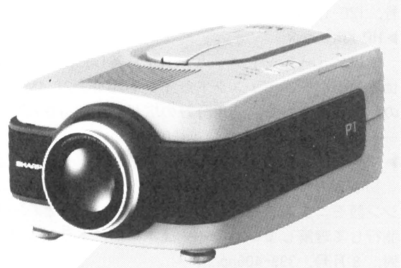
価格は8,5000円（税別）となっている。

〈問い合わせ先〉

㈱ログ ☎03(3837)2595

コンパクト液晶ビジョン
XV-P1
シャープ

XV-P1



シャープは，コンパクト液晶ビジョン「XV-P1」を発売した。

本機は，新開発の30万画素高精細液晶パネル1枚を搭載した「単板方式」を採用。これにより，従来機の約3分の1に小型化，軽量化することができた。さらに，明るさの均一性を高めた高輝度映像を実現している。

さらに，アンプ，スピーカーを内蔵（モノラル）しているため，ビデオデッキやビデオカメラと接続するだけで大画面を楽しめるようになっている。

また，“ハイブライトスクリーン”(別売)と組み合わせることにより，明るい場所での視聴もできる。

なお，ビデオ入力は2系統，S映像入力端子は1系統装備している。入力はビデオ入力1にS映像入力が優先される。

価格は220,000円（税別）となっている。

〈問い合わせ先〉

シャープ㈱☎03(3260)1161,06(621)1221

ビジネスパーソナルファクシミリ
UX-11
シャープ

UX-11

シャープは，留守番電話とファクシミリを一体化させたビジネスパーソナルファクシミリ「UX-11」を発売した。

ファクシミリ機能としては，A4標準原稿なら12秒で伝送可能な「高速伝送機能」や最高5枚まで連続して送信できる「自動給紙機能」を装備。そして，細かな図面などの送信に適した「精細（セミスーパーファイン）モード」と，ハーフトーンを美しく再現する16階調の「中間調モード」も搭載している。

留守番電話機能としては，留守ボタンを押すだけで留守録モードになる「ワンタッチ留守番電話機能」,留守録を指定の電話に転送できる「指定先転送機能」や電話の着信をポケットベルに知らせてくれる「ポケベル転送機能」がある。もちろん外出先のプッシュホンから留守録機能を操作することも可能。

また，別売のハンドスキャナ「UX-05HS」を接続することにより，ノートなどの厚みのある原稿や新聞のように大きな原稿なども，直接ファクシミリで送信することもできるようになる。

価格は112,000円(税別)となっている。

＜問い合わせ先＞

シャープ㈱☎03(3260)1161,06(621)1221

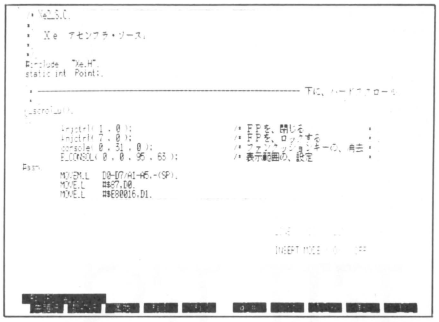
通信機能付きスクリーンエディタ
Xe ver.2
エル・クラフト

エル・クラフトは，X68000用スクリーンエディタ「Xe ver.2」を発売した。

「Xe ver.2」は1行95文字，3000行までのテキストを扱える通信機能付き学習用シングルファイルスクリーンエディタである。C言語によるソースリストが公開されており，ユーザーは自分の手でエディタの改造などを自由に行えるようになっている。

なお，同ディスクには簡易ファイルセレクタ「FS.X」も収録されている（こちらもソースリスト付き）。

価格は9,800円（税別）となっている。



〈問い合わせ先〉

㈱エル・クラフト　☎0559(71)2015

INFORMATION

ダイヤルQ²アスキー書籍・データライブラリー
アスキー

アスキーは，ダイヤルQ²サービスを利用してパソコン通信で提供する「ダイヤルQ²アスキー・書籍データライブラリー」を開始した。

このサービスを利用することにより，ユーザーは提供された書籍の中のデータ，サンプルプログラムを自由に入手することができる。そして，書籍の一部分から1冊全体のデータまで，ユーザーが必要としているデータを選択して入手可能。

今回提供される書籍は，以下の9タイトル。「花子入門」「Multiplan 3.1」「MIFESを256倍使うための本」「VZを256倍使うための本」「X68000 パワーアッププログラミング」「Cプログラムブック I」「Cプログラムブック II」「Cプログラムブック III」「応用C言語」

なお，入会手続きは不要。

- アクセス番号：☎0990-343260
- アクセス料：90円/分（通話料別）
- 通信プロトコル：Xmodem,Ymodem
- 通信ボーレート：2400bps

〈問い合わせ先〉

㈱アスキー　☎03(3797)3225

FILES Oh!X

このインデックスは，タイトル，注記——著者名，誌名，月号，ページで構成されています。暦のうえではもう秋だけど，まだまだ暑さは続きます。夏バテしないように，しっかり栄養をつけて，たまには運動もしましょうね。

参考文献

I/O　工学社
ASCII　アスキー
コンプティーク　角川書店
テクノポリス　徳間書店
天文ガイド　誠文堂新光社
トランジスタ技術　CQ出版社
POPCOM　小学館
マイコンBASIC Magazine　電波新聞社
マイコン　電波新聞社
LOGIN　アスキー

一般

▶THE NEWS FILE
'92東京おもちゃショウ会場の模様をレポート。ゲームマシンはもちろん，子供たちの間で大人気のバーコードバトラーの新機種など，各社ともハイテク電子技術を駆使した話題の商品を出展。——編集部，LOGIN，14号，38-39pp.

▶アルゴリズムを見切ったぞ!?
3Dゲームの巻・その３。3D座標の計算アルゴリズムについて，X68000のX-BASICのリストなどをサンプルにやさしく解説。——おにおん，テクノポリス，8月号，132-136pp.

▶ど〜するど〜なる!?　パソコンゲーム！
1991年度のパソコンゲームソフトの売り上げは，前年と比べると20％の落ち込みだった。着実に人気を伸ばしつつあるゲーム専用機と，マニア向けで難しく見られがちなパソコンを比較しつつ，パソコンゲームの未来を考える。——編集部，テクノポリス，8月号，137-140pp.

▶THE NEWS FILE
上野と新宿に開店したコンセプトゲームセンター，「パセラ」「タイトー・イン・ゲームワールド」の紹介。ビジネスショウはペン型入力デバイスが話題。シャープの提案する次世代電子情報「ハイパー電子マネージメント手帳」など。——編集部，LOGIN，13号，36-43pp.

▶電脳通信
各社の最新発売機種を紹介。シャープの「書院パソコン」は，ワープロ「書院」と「DOS/V」パソコンが一体化された新しいスタイルのマシンだ。——編集部，コンプティーク，8月号，237p.

▶ワープロ/パソコン通信新聞
ワープロでもできる戦闘ゲームがNIFTY-Serveにオープン。子供のためのフォーラム，第１回フリーソフトウェア大賞の受賞作など，パソコン通信の最新ニュースを掲載。——山本まさこ，マイコンBASIC Magazine，8月号，274-277pp.

▶最新機種　今買うならこれだっ
ペンを使った手書き入力のできるシャープの電子手帳「PV-F1」をはじめ，ペンコンピュータ，PC-9801など今年の春夏の新製品を紹介し，業界の動向を読む。——編集部，ASCII，8月号，185-207pp.

▶得するパソコン活用術
パソコンのランニングコストを抑える方法を伝授する。X68000をプログラミングマシンとして安価に使うための知恵を紹介し，安くてちょっと遊べるマシンとしてX1 turboZも登場している。——編集部，ASCII，8月号，225-240pp.

▶Digi-Ana Valley
デジタルで録音してデジタルで聴く。ソニーが開発したMDは音質・ポータビリティなどを高いレベルで両立させ次世代オーディオシステムだ。その仕組みと音声データ圧縮技術の深淵に迫る。——編集部，ASCII，8月号，289-296pp.

▶欧州ハイテク事情
ミラノで行われたMACWORLD EXPOSITIONの模様を伝える。イタリアでのMacの普及はまだまだこれからということもあって，啓蒙的なブースが多かったほか，美術系の出版社が出展していたのが印象的だったとか。——菊地薫，ASCII，8月号，352-353pp.

▶News Room
アップルが先頃発表し，シャープが製造に参加することが話題を呼んだPDAに対して所見を述べる。——山田憲一，マイコン，8月号，100-103pp.

▶入門DIY工作
簡易ディスプレイ切り替え器を製作する。ディスプレイの信号線の解説を行い，製作上のアドバイスと使用法を説明する。——石川至知，マイコン，8月号，234-235pp.

▶'92東京おもちゃショウ
６月４日から７日までの４日間，千葉市の幕張メッセで行われた東京おもちゃショウの模様をレポートし，現在のおもちゃの傾向を対談形式で分析する。——あゆさわかつみ他，マイコン，8月号，273-281pp.

▶ページプリンタの基礎知識
DTPの普及とともに関心の高まっているページプリンタ。そのメカニズムや記述言語の使い方を解説し，インクジェットのメカニズムにも言及。最新のページプリンタカタログとともにお届けする。——川田徹他，I/O，8月号，54-73pp.

▶印刷終了通知ブザー
プリンタの印刷が終わると鳴るブザーのハード工作。外部電源も不要の優れモノだ。——森羅万象，I/O，8月号，120-123pp.

▶HP Kittyhawk
横川ヒューレットパッカードが発表した，1.3インチの世界最小ハードディスクドライブを紹介。21.4Mバイトの記憶容量をもち，応用分野の広がりが期待されている。——編集部，I/O，8月号，160-161pp.

▶マシン語モニタを作ってハードとソフトを理解しよう
初心者のためのZ80マイコン教室。AK-Z80を題材に，マシン語モニタを組み込むことによってソフトとハードを並行して理解していく。——茂木東樹，トランジスタ技術，8月号，399-406pp.

MZシリーズ

MZ-1500(BASIC 5Z-001)

▶ベーマガ・パニック
けんかしてるヤツはぶっとばせ。上から落ちてくるベーマガ編集部の人たちを操作して，けんかをさせ，ふっとばす。アクションパズルゲーム。——Blues Power，マイコンBASIC Magazine，8月号，124-125pp.

MZ-2500(BASIC-M25)

▶TRIANGLE
画面の黄色以外の３色の三角形をすべて消す。色合わせパズルゲーム。——工藤俊介，マイコンBASIC Magazine，8月号，126-128pp.

X1/turbo/Z

X1シリーズ

▶THOUGHT
コンピュータとディスプレイをつなげ！　アクションパズルゲーム。——山根幸一郎，マイコンBASIC Magazine，8月号，153-154pp.

▶誌上公開質問状
X1Gモデル30がつながるX68000用ディスプレイは？という質問に答える。——多田太郎，マイコンBASIC Magazine，8月号，178p.

X1＋FM音源ボード（要NEW FM音源ドライバ）

▶ストリートファイターII 〜ケンのテーマ〜
カプコンのゲームミュージックプログラム。——川村賢治，マイコンBASIC Magazine，8月号，173-174pp.

X1turboシリーズ

▶移植版おむすび探偵団
どんなおむすびの中身もおまかせ！　中身がわからなくなってしまった大量のおむすびを前に，おむすび探偵団のリーダーとなって中身を当てるゲーム。——舟生日出男，マイコンBASIC Magazine，8月号，155-156pp.

X68000

▶X68000新聞
完成を間近に控えた「ポピュラスII」を紹介。国産マシンの移植はX68000がいちばん最初だ。X68000 CompactXVIの発売で少々混乱気味のフロッピーディスクドライブ。ついにシャープ純正の５インチディスクドライブが発売された。そのほか，計測技研より発売されるCD-ROMドライブやX68000 CompactXVIに内蔵する80Mバイトのハードディスクを紹介。新着ゲームのほうは「餓狼伝説」，PC-9801版で大人気のアドベンチャー「ふしぎの海のナディア」がX68000に移植決定の話題など。——編集部，LOGIN，13号，258-261pp.

▶NEW SOFTWARE
ビジネスにも使える本格的グラフ＆チャートソフト「CHART PRO-68K」を紹介。各種データベースで作成したデータをもとに，多彩なグラフを作成するアプリケーション

ソフト。――編集部，マイコンBASIC Magazine，8月号，88-89pp.

▶DRAGON PANIC

襲いかかるドラゴンから逃げろ。先に喰われてしまったら負け。2人用アクションゲーム。――林純一，マイコンBASIC Magazine，8月号，157-158pp.

▶エキサイティング・ラーメン

ラーメン屋とお客との戦い。ひたすら客の注文どおりにラーメンを作り続ける。記憶力アクションゲーム。――石川潤，マイコンBASIC Magazine，8月号，159-160pp.

▶RUIN

誰も手にすることのできなかった宝石を手に入れるために遺跡に足を踏み入れた……。「ゼルダの伝説」風アクションRPG。――AHO，マイコンBASIC Magazine，8月号，161-163pp.

▶ストリートファイターII　～タイトルBGM～

カプコンのゲームミュージックプログラム。要NAGDRV。――荒木潤，マイコンBASIC Magazine，8月号，175-176pp.

▶誌上公開質問状

CM-32LはX68000につながるか？　X68000用として発売されているディスプレイにPC-9801シリーズをつなぐことはできるか？　X68000のマシン語を覚えたいが，いい参考書はないか？　などの質問に回答。――多田太郎，マイコンBASIC Magazine，8月号，177-178pp.

▶SOFT EXPRESS

幅広いファンをもつポピュラスの第2弾「ポピュラスII」がX68000に移植。8月末の発売を待て！　そのほか初心者からマニアまで興奮のF1ゲーム「OVERTAKE」の紹介。機種別新着ソフトのリストなど。――編集部，コンプティーク，8月号，62-67pp.

▶Hardware Hot Press

各社の新発売機種を紹介。X68000CompactXVIに最適なシャープ純正5インチフロッピーディスクドライブ「CZ-6FD5」など。――編集部，POPCOM，8月号，24-25pp.

▶最新SLG徹底比較

最新シミュレーションゲームを紹介。X68000代表は，当然話題の「ポピュラスII」だ。――編集部，POPCOM，8月号，66-67pp.

▶ゲームの達人

格闘ゲームの老舗がX68000に完全移植！　カプコンの「ファイナルファイト」を紹介。――相撲レスラー本田，POPCOM，8月号，100-101pp.

▶GAMING WORLD

最新ゲームの紹介。手に汗握る3Dバトルシミュレーション「バトルテック～奪われた聖杯～」，発売予定の「エアーマネジメント」「キャッスルズ」。――編集部，テクノポリス，8月号，18-37pp.

▶SUPER NEW SOFT

あのカプコンが，ついにパソコンゲーム界に進出だ。その第1弾としてX68000の「ファイナルファイト」が選ばれた。ファン待望のアクションゲームを紹介。――編集部，LOGIN，14号，14-15pp.

▶X68000新聞

スプラッタホラー「デッド・オブ・ザ・ブレイン」は，なんとあのフェアリーテールの作品。そのほか新着ゲーム「三國志III」。SX-WINDOW上で動く「SOUND SX-68K」「COMMUNICATION SX-68K」など，シャープのSX-WINDOWに対する意気込みが伝わってきそうな新製品たちを紹介する。――編集部，LOGIN，14号，222-225pp.

▶AV STRASSE

計測技研から発売されたCD-ROMドライブ「KGU-XCD」を取り上げ，SX-WINDOW上のアプリケーションの使い勝手やソフトウェア供給予定も併せて紹介する。また，SOUND SX-68Kのベータバージョンも登場。――編集部，ASCII，8月号，301-304pp.

▶美瑛

北海道美瑛町の町おこしと連動したパズルゲーム「美瑛」を取り上げる。スタイルはテトリスタイプのパズルゲームだが，前田真三氏による美瑛町の写真が取り込まれているのが特徴。――猪野清秀，マイコン，8月号，126p.

▶なんでもQ＆A

Human68kの起動時にRAMディスクを確実に確保する方法，SX-WINDOW ver.2.0でツァイトの「書体倶楽部」を使用する方法などを聞く。――シャープ株式会社AVCシステム事業推進室，マイコン，8月号，252-253pp.

▶DUAL TANK

2人用対戦型戦車ゲーム。C言語を勉強中の人のための習作でもある。プログラムは付録ディスクにも収録。――土方嘉徳，I/O，8月号，95-99pp.

▶X68000用MIDIインタフェイスボードの製作

純正MIDIインタフェイスボードと同じように使えることを目標にして製作するハード工作。――奥村暢朗，トランジスタ技術，8月号，379-381pp.

▶ST-4による天体画像

X68000を使った天体用CCDカメラの活用例を報告。撮影した画像をソフトウェアで処理し，さらにX68000のZ's STAFF PRO-68Kを使って画像の仕上げを行う。――福島英雄，天文ガイド，8月号，71-74pp.

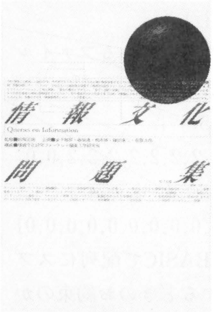
ポケコン

PC-E500

▶The Baseball

ひとりでも2人でも楽しめてしまう，カードゲーム風対戦野球ゲーム。――津崎直，マイコンBASIC Magazine，8月号，165p.

新刊書案内



情報文化問題集
松岡正剛監修
情報文化研究フォーラム＋編集工学研究所構成
☎03(5484)4060
NTT出版刊
A4判　371ページ
3,200円(税込)

NTT出版の情報文化研究フォーラムシリーズ最新作がこの「情報文化問題集」である。問題集といっても質問が並んでいるわけではもちろんなくて，情報文化で問題とされている部分を集めた，と捉えるのがいい。情報文化研究フォーラムなどという面妖なフォーラムは何をするところか，というと，「メディアとコミュニケーション」「AIと言語」「遊び・芸能・宗教」「都市の歴史とデザイン」および「こどもと情報環境」の5ジャンルにおいて，それぞれ平均10回ずつの討論を繰り広げるもの。各ジャンルにはコーディネーターがいて，フォーラムに呼ばれた各ジャンル専門家の出席者がいる。1989年にそのフォーラムで論議された内容をまとめたのが本書。テーマに沿って，まず出席者によるテーマの概論，続いて対談，という構造がずらりと並んでいる。現在，これだけのメンバーを一堂に会した本はないというだけでも価値がある。1989年だからといって古いと思ってはいけない。1992年でも十分に通用する内容であるし，1989年という年に縛られてしまうようなメンバーはいないからだ。

本書が追求するのは，コンピュータにとっての情報ではなく，技術，とりわけコンピュータが発達していく時代での人間にとっての情報であり，そこではメディア，コミュニケーション，脳，文化など，テクニカルな側面だけでは捉えきれないジャンルに対してテクノロジーを踏まえたアプローチをすることが要求されている。どちらか一方の視点では必ず破綻する問題であり，双方を同じレベルの視点をもって見つめられる人が必要とされているのだ。本書はそういう意味で非常に面白い。各論が短いため，詳細は各テーマの専門書を必要とするが，おかげでとっつきやすくなっており，興味のあるジャンルを拾い読みするだけなら気楽に取り掛かれる。NTT出版のシリーズではいちばんお得な本ではないだろうか。　(K)

遊園地の現在学
松本孝幸著
JICC出版局刊
☎03(3234)3688
新書判　147ページ
980円(税込)

最近，遊園地のあり方が変わってきた。以前は大きな敷地をもつ郊外型が主であったし，それほど大掛かりな仕掛けがあるとは思えなかった。が，ここ数年バーチャルリアリティなど最新テクノロジーを駆使したものがどこの遊園地にもお目見えしているし，街の中にもそういった技術だけを寄せ集めた小規模遊園地（いわゆるアミューズメントスポット）も登場し始めている。

本書は，とくに遊園地のジェットコースターなどを例に，最新技術によって得られる人工現実感，そして未知なる超都市や超感覚などを探っていくものである。

電脳筒井線　―朝のガスパール・セッションPART2
筒井康隆著
朝日新聞社刊
☎03(3545)0131
四六判　254ページ
800円(税込)

昨年発売された同名タイトルの続編。本書は，朝日新聞に連載されていた長編小説「朝のガスパール」と連動して，パソコン通信のASAHIパソコンネット上で繰り広げられた会議の記録をまとめたものだ。パソコン通信上だから，もちろん多数の参加者がいて，「朝のガスパール」についての意見のみならず，いろいろな話が織り交ぜてあり，なかなか楽しく読める。もちろん，「朝のガスパール」を読んでいればより楽しめるが，パソコン通信の特徴をそのまま生かして本にしているので，パソコン通信とはどういうものかを知りたい人などにもうってつけの本である。

創刊10周年記念PRO-68Kに収録されたSM.Xは手軽にスプライトデータの制作ができそうなのですが，SM.Xでセーブしたスプライトデータを X-BASICやアセンブラで利用する方法がわからないので困っています。そのための方法を教えてください。

神奈川県　堀江　良靖

同様の質問が多数送られてきました。SM.Xで保存されるスプライトデータ（＊.SP）やパレットブロックデータ（＊.PAL）は，SM.X独自のデータ構造で保存されるため，マシン語プログラムならともかく，BASICでは扱いにくくなっています。そのデータ構造は6月号の66ページに公開されています。それを参考にしてスプライトデータとパレットブロックデータを読み出して，X-BASICで利用できるように変換するプログラムを作成してみました。

プログラムはX-BASICで書かれています。リスト1が＊.SPのファイルからスプライトデータを読み出すプログラム，リスト2が＊.PALのファイルからパレットブロックデータを読み出すプログラムです。

どちらとも読み出したデータは，X-BASICで利用できるかたちにしてファイルに書き出します。入出力ドライブおよびファイル名は，各自自分の環境にあわせて変更してください(30，40行)。

リスト1を実行して出力されるファイルの内容は，

```
dim char sp1(255)= {
+0,0,2,2,2,2,2,2,2,2,2,2,2,2,0,0,
                ⋮
+0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0}
```

といったように，X-BASICで配列にスプライトデータを定義するときのお約束のかたちになっています（配列の内容は説明のために書いたもので，同じ内容でなければならないということはありません）。

もし，n個のスプライトデータがあれば，配列名がsp2，sp3，sp4……spnと続いていきます。このファイルには行番号がついていないので，X-BASICのLOAD@コマンドを使ってロード（アペンド）してください。

実際のスプライトパターンの定義にはspdef命令を使います。たとえば配列sp1のスプライトデータをパターンコード0に定義するなら，

```
sp_def(0,sp1,1)
```

となります。詳しくはX-BASICのマニュアルを参照してください。

リスト2を実行して出力されるファイルは，

```
dim int pal1(15) = {
+0,0,……,65534}
```

のようになっています（繰り返しますが，配列の内容は説明のために書いたもので，

リスト1

```
 10 int ai
 20 dim char data(128)
 30 ai=fopen("h:test.sp","r")
 40 bi=fopen("h:sp_data","c")
 50 i=1
 60 while feof(ai)<>-1
 70   fread(data,128,ai)
 80   sprite_data(i)
 90   i=i+1
100 endwhile
110 fputc(&H1A,bi)     /* ﾃｷｽﾄEOFｺｰﾄﾞ
120 fcloseall()
130 end
140 func sprite_data(sp_num;int)
150 int i,j,k
160 fwrites("dim char sp"+str$(sp_num)+"(255)={",bi)
170 crlf()
180   for j=0 to 15
190     for k=0 to 3
200       sprite_data_write(i,k)
210     next
220     for k=64 to 67
230       sprite_data_write(i,k)
240     next
250     i=i+4
260     if i<=60 then { fputc(asc(","),bi)
270         } else { fputc(asc("}"),bi) }
280     crlf()
290   next
300 endfunc
310 /*
320 /* 改行コードを出力
330 /*
340 func crlf()
350   fputc(13,bi)
360   fputc(10,bi)
370 endfunc
380 /*
390 /* ｽﾌﾟﾗｲﾄﾃﾞｰﾀを出力
400 /*
410 func sprite_data_write(i;int,k;int)
420   if k=0 then { fputc(asc("+"),bi)
430       } else { fputc(asc(","),bi) }
440   fwrites(str$(data(i+k)/16),bi)
450   fputc(asc(","),bi)
460   fwrites(str$(data(i+k) and 15),bi)
470 endfunc
```

リスト2

```
 10 int ai
 20 dim char pal_data(32)
 30 ai=fopen("h:palet.pal","r")
 40 bi=fopen("h:palet_data","c")
 50 i=1
 60 while feof(ai)<>-1
 70   fread(pal_data,32,ai)
 80   palet_data(i)
 90   i=i+1
100 endwhile
110 fputc(&H1A,bi)     /* ﾃｷｽﾄEOFｺｰﾄﾞ
120 fcloseall()
130 end
140 func palet_data(palet_num;int)
150 int i,j
160 i=0
170 fwrites("dim int pal"+str$(palet_num)+"(15)={",bi)
180 crlf()
190 for j=0 to 15
200   if i=0 then fputc(asc("+"),bi)
210   fwrites(str$(pal_data(i)*256+pal_data(i+1)),bi)
220   i=i+2
230   if i<32 then { fputc(asc(","),bi)
240         } else { fputc(asc("}"),bi) }
250 next
260 crlf()
270 endfunc
280 /*
290 /* 改行コードを出力
300 /*
310 func crlf()
320   fputc(13,bi)
330   fputc(10,bi)
340 endfunc
```

表1　アセンブラ用変更点

```
リスト1 sp_def.bas の変更点
  160 fwrites("pat"+str$(sp_num)+":",bi)
  410   if k=0 then { fwrites(" dc.b ",bi)
  260,270行削除

リスト2 sp_color.bas の変更点
  170 fwrites("pb"+str$(palet_num)+":",bi)
  200   if i=0 then fwrites(" dc.w ",bi)
  230   if i<32 then fputc(asc(","),bi)  240行削除
```

同じ内容でなければならないということはありません)。*.PALにすべてのパレットブロックをセーブしたなら，Pal1～Pal15までの配列がパレットブロック1～15に対応します。

また，配列の要素番号がパレットコードに対応します。このファイルにも行番号がついていないので，LOAD@コマンドを使ってロードしてください。

たとえば，pb1の内容をパレットブロック1に設定するなら，

```
for i=0 to 15
    sp_color(i,Pal1(i),1)
next
```

のようにします。

また，読み出しプログラムを使わず，直接マシン語プログラム中で利用したいという方のためのリストの変更点を図1に紹介します。これでアセンブラソースリストにDC命令でデータが出力されるようになります。あとは，各自でこの形式にあったスプライト定義ルーチンを作って呼び出すようにしてやればいいわけです。

.SPに保存されているスプライトデータはキャラクタコードの若い順に並んでいます。またCANVASのデータである.PATは，先頭の1ワードがスプライトデータの横のサイズ，続く1ワードが縦のサイズを表し，その後ろにスプライトデータが格納されています。

リスト1は*.PATのデータ構造に対応していませんので，少しプログラムを組める方ならば入力ファイルの拡張子に応じて，先頭の2ワードを特別扱いするようにプログラムを改造してもいいでしょう。もっとも，*.SPで保存すればことは足りますので，わざわざそうする必要はないと思いますが。

(影山　裕昭)

先日，友達からPC-9801で作ったテキストファイルなどをもらったのですが，半角片仮名のファイル名のもので読み込めないものがありました。マニュアルによればHuman68kでも片仮名のファイル名は許されているはずですがなぜでしょうか。ちょっと気になるのは，片仮名のファイルがすべてダメというわけでもなく，一部のものだけが読み込めないことです。X68000とPC-9801では一部のコードが違うのでしょうか。また，どうしても読み込むことはできないのでしょうか。

富山県　藤代　俊己

おそらく読み込めなかったファイルは，音引きとしてマイナス記号を使っているものと思われます。MS-DOSではファイル名の一部に“－”を使うことができますが，Human68kでは禁止されているのです。

このようなものはディスクの管理情報を直接書き換える，MS-DOSマシンでリネームする，フリーソフトウェアのtwentyone.xなどのようなHuman68k拡張ツールを使用する，というのが対処法として知られています。

しかし，最新のHuman68k ver.2.03ではマイナス記号をファイル名に使用できるように拡張されていますので，システムディスクのHuman68kを新しいものに取り替えるものもよいでしょう（ただしマニュアルには明記されていません)。新しいHuman68kはX68000CompactXVIのシステムディスク，XC ver.2.1，SX-WINDOW ver.2.0などのパッケージに使用されています。

幸い，Human68kの大きさが，

ver.2.01 54174バイト
ver.2.02 54240バイト

なのに対し，

ver.2.03 53626バイト

と少し小さくなっていますから，これらのバージョンからのシステムの差し替えは簡単です。ここではハードディスクがAドライブ，フロッピーがDドライブだと仮定して手順を解説しましょう。

まず，コマンドモードでAドライブのルートディレクトリに移動し，

```
A>ATTRIB -S HUMAN.SYS
```

を実行します。これでHUMAN.SYSが可視状態になったはずです。次に，

```
A>DEL HUMAN.SYS
```

を実行します。システムディスクのHuman68kが消去されました。ここからはすべての作業が終わるまで絶対にAドライブに新しいファイルを作成しないでください。もちろんリセットもいけません。

SX-WINDOW ver.2.0のマスターディスクなどをDドライブに入れ，

```
A>D:
```

でカレントディレクトリをDドライブに移動します。

```
D>SYS A:
```

でHuman68kが連続した領域に転送されたら操作は終了です。

このように新しいシステムが従来よりも小さい場合，本来ならSYSコマンドだけでシステムの転送ができるはずなのですが，スタッフの話によると稀に連続したFATへのシステム転送が失敗することがあるようですので，一度HUMAN.SYSを削除しておいたほうが確実でしょう。

万一，失敗した場合は最初からやり直すか，同じ手順で元と同じバージョンのHuman68kを転送してください（おそらくは途中でAドライブに新規ファイルを作ってしまったか，Human68kがver.1の場合)。復元が不可能になっても他のファイルは無事ですので，その時点でファイルのバックアップを行えば最悪の事態は回避できるはずです。リセットしてしまった場合はあわてずにフロッピーから立ち上げ直してください。

なお，マニュアルに使用可能とされている文字以外にも“$”，“‘”（シフト＋@キー)，“%”がファイル名として使用できるようです（従来バージョンより)。ただし，“%1”などというファイル名を作成した場合はバッチ処理などで不都合がある可能性もありますので使用しないほうがよいでしょう。また，当然のことながら，“－”はオプション指定とみなされますので，ファイル名の先頭文字には使用できません。

(中野　修一)

質問にお答えします

日ごろ疑問に思っていること，どんなことでも結構です。どんどんお便りください。難問，奇問，編集室が総力を挙げてお答えいたします。ただし，お寄せいただいているものの中には，マニュアルを読めばすぐに解答が得られるようなものも多々あります。最低限，マニュアルは熟読しておきましょう。質問はなるべく具体的に機種名，システム構成，必要なら図も入れてこと細かに書いてください。また，返信用切手同封の質問をよく受けますが，原則として，質問には本誌上でお答えすることになっていますのでご了承ください。なお，質問の内容について，直接問い合わせることもありますので，電話番号も明記してくださいね。

宛先：〒108 東京都港区高輪2-19-13
NS高輪ビル
ソフトバンク株式会社出版部
「Oh! X質問箱」係

STUDIO X ON AIR

FROM READERS TO THE EDITOR

すでに海はクラゲだらけ。華やかな夏が過ぎようとしています。夏の思いが実現した人にも，見事砕け散った人にも共通に季節は移り変わるでしょう。残暑を感じながら，ちょっとだけ思い出に浸るのもいいんじゃないかな。

◆DōGA CGAシステム。予想どおりのすばらしい出来。バックナンバーを引っ張り出し，マニュアルなしながらも結構遊んでいます。

中嶋　祥史(24)神奈川県

◆おおっ！　DōGA CGAシステムだ。先月からとても楽しみにしていたので，さっそく立ち上げちゃる。なになに，お試しディスク？　モーションデータ集みたいなやつか。2，3回試しにアニメーションを作ってみると，やっぱり自分でもなにか作りたくなりますね……。なにっ，マニュアルは手に入れるのに1カ月かかるんですか。とりあえず，昔のOh!Xを読み返しながらなんとか……なりそうもないですね。早くマニュアルきておくれ〜。

小林　宏教(18)新潟県

◆かねてからDōGA CGAシステムを入手したいと思っていたのですが，なかなかチャンスがありませんでした。今回の付録ディスクにはとても感謝しています。

堤　雅秀(24)神奈川県

◆DōGA CGAシステムのお試しディスクは，受験生に優しいシステムでした。マウスでチョイと操作して，X68000が計算している間に受験勉強。疲れてきた頃に計算完了。できたアニメーションを見て疲れをふっ飛ばし，また机に向かう。まったくありがたいシステムです。DōGAさんありがとう。

安藤　哲(18)山形県

大反響のDōGA CGAシステム。がんばって使いこなしてくださいね。

◆7月号の特集「TORNADOの秘話」には，非常に感銘を受けました。思わず「よーし，おれもやってやるぜ！　次回のCGAコンテストのグランプリはこのおれがもらったあ」と叫んでしまったほどです。

青島　一高(23)静岡県

◆7月号の文月氏の記事を読んで，2輪のロードレースに出ていた頃の自分を思い出しました。どんなジャンルにいる人でもなにかを成し遂げるには，自分が納得するまで「あきらめない」ことがやっぱり大切。私はそんなTORNADOを見てみたい。

上田　剛(33)北海道

文月氏の熱い情熱には見習うべきところがあるでしょう。読者の皆さんも負けてなんかいられませんよ。

◆お試しディスクのおかげで，ここ2，3日X68000の電源は入れっぱなし。初めは軽い気持ちで「ワゴン」を画質「悪い」でやってみて（待っている間はもちろんスーパーファミコンのストIIね）思わず感動しました。このハガキを書いているいまも「F-15」のアニメーションを作っています。ところで何日ぐらいX68000の電源を入れっぱなしにしても大丈夫なんでしょうか。

藤木　健二(16)東京都

本体についてはかなりの期間電源を入れっぱなし（1，2カ月ぐらい）にしていても平気です。ただ，ディスプレイはブラウン管を傷める可能性があるので，こまめに消しておきましょう。

◆DōGAはすばらしいです。お試しディスクで作ったサンプルを用いていろいろ遊んでいますが，これで付録＋3,000円は安すぎます。といってもまだ入金していないけど，ボーナスが出たら入金しよう。現在，マニュアルなしでもモデリングから動画合成などをやって楽しんでいます。ところで，高橋さんがイラストにこだわっていて非常に嬉しい。で，イラスト増ページは予定していませんか。

見浦　崇(23)長野県

実現するためには読者の皆さんの協力が不可欠。実力でイラストコーナーを実現しましょう。挑戦，待ってます。

◆「連載のすべて」を読んで「ああ，10年前から続いている連載はもうないのか」と思いながら，1982年のOh!MZを読んでいました。そこで気がついたのですが，題名も変わらずいまでも続いている連載(?)があるじゃないですか。その名は「ぼくらの掲示板」と「SHIFT BREAK」。

石井　一成(22)山梨県

これはすごい，と喜ぶべきことなんだろうか。ちょっとだけ悩んでしまいました。

◆ペンギン情報コーナーのX68000用CD-ROMドライブ「KGU-XCD」を見たときには，とても嬉しかった。やっとX68000にもCD-ROMドライブを接続できる。同時発売の「フリーウェアセレクション」にも注目をしてしまいますね。ぜひ，レビューをやってほしいところです。

岡島　昌之(28)大阪府

ごめんなさい。8月号はおろか9月号でも間に合いませんでした。来月こそレビューをしますので楽しみにしていてください。

◆7月号105ページを読んで僕が思った感想を書きます。僕はいつも村田さんの記事をとても楽しみにしています。なぜなら，X68000のアセンブリ言語が，わかりやすく丁寧に解説されているからです。僕がアセンブラをやってみようと思ったのは，村田さんの記事を読み「これくらいなら僕にも書けるかもしれない」と感じたからです。村田さん，これからもグレイトでありつづけてください。

今井　雅治(19)岐阜県

今井さんもグレイトになるためにがんばってくださいね。

◆最近，汚れていたキーボードを分解してキートップとカバーを水洗いしてみました。いやあ，とてもきれいになりました。新品同様です。と喜んでいたのですが……あれ？　F6キーがおかしいぞ。

井戸　滋(18)鳥取県

ここにまた，キーボード分解掃除の犠牲者が出てしまいましたか。面倒ですがキートップを分解せずにガラスクリーナーなどでセコセコ拭き，ほこりは掃除機で吸い出すのがいちばん安全です。

◀住友　智代　愛媛県
帰りたくても帰れない，月が恋しいウサギ娘，なのかな。ころころしたキャラクターが竹本泉みたいでかわいいですよ。

◀日高　光代　宮崎県
2Dペイントソフトなら「Z'sSTAFF ver 3.0」「MATIER」，この2本がお勧め。両方あれば最高ですが「Z'sSTAFF」＋「Z's-EX」でも十分でしょう。

◆PCM8やS-OSエミュレータのソースプログラムを見て思ったんですが，最近のアセンブラプログラムの流行は「怒濤のテーブル化」のようですね。メモリ需要が高まるわけだ。

清水　了(18)大阪府

いわゆる力技のメモリ展開。プログラムの高速化のテクニックとしては，かなり昔から使われていたものです。

◆X68000 XVIを購入してはや半年。購入してから1，2カ月間は，X-BASICを立ち上げるにしてもSX-WINDOW上からしかできず，プログラムを作るにしてもたいしたことはおろか，まともに作ることすらできませんでした。そんな私がいまでは，そこそこの大きさのプログラムを作れるようになりました。これも私のようなユーザーでも温かく見守ってくれた貴誌のおかげです。今度は，CGかアセンブラでもやろうかな，と思っています。これからもよい誌面を期待しています。　高三　直也(16)愛媛県

それは，がんばってついてきた高三さんの努力もひと役かっています。どんどん自信をつけていってください。

◆質実剛健と書いて「びんぼう」と読む。さて，「10年分のSTUDIO MZ・X」を読んで，ペットの餌用にパンの耳をもらっていくおばさんと，競い合ったことを思い出しました。最近はちゃんと米の飯を食べています。ちなみに今年の誕生日，友人からのプレゼントは「パンの耳」でした。

中村　学(21)石川県

今度は，気前よくその友人におごってあげましょう。貧乏にならない程度にね。

◆学生時代にはX1を使っていた。ソフト会社(某国民機メーカー系)に入ってはや7年目。すでにX1は入社時に友人にあげてしまい，家でコンピュータを触ることは二度とないだろうと思っていました。でもやっぱり，「Oh!MZ」やX1にドキドキしていたあの感覚が忘れられなくて，先月ついにX68000を買った。会社で非国民といわれようとも，最高にドキドキさせてくれるこのマシンが大好きになりそうだ。

平野　輝元(28)兵庫県

自分の好きなものなんですから，恥じることなく堂々と使えばいいのです。

◆私：あ～X68000欲しいなあ
友人：なにそれ？
私：パソコンだよ
友人：ああ，それってすげえ電話代がかかるんだろ？
この会話でもわかるとおり，友人はパソコンとは通信するだけのものだと思っていたようです。一般人のパソコンに対する認識ってこんなものなんでしょうか。　杉山　洋之(19)埼玉県

要するに自分が興味を持っていないものに対しては，それほど追求しないだけじゃないのかな。

◆「CZ-6BE2A」の取扱説明書中「お客様が取り付けたのち，故障，異常が生じたときは責任を負いかねます」と記載されていた。しかし，不思議なことにセッティング手順が説明書に明記されている。なにか矛盾していないだろうか。

増田　秀樹(26)東京都

一応，X68000は家電ということなんで，そのへんの規則がからんでくるため矛盾が生じるんでしょう。

▲鈴川　美佳子　東京都
ナムコワンダーエッグの「ギャラクシアン³」と，GEOPOLISにあったセガの「ヴァーチャルレーシング」はお勧め。ひまがあったら遊びに行こう。

▲岩瀬　喜代美　福岡県
夏らしい涼しげなイラスト。ずいぶん調子が悪い，といっていますけど，スランプなんかふっ飛ばすくらい元気に遊びましょう。

◆次々と増殖する各種データにまみれ，さらにDōGA CGAシステムまで入手してしまい。本格的にハードディスクがほしくなった今日この頃。創刊10周年特別企画を楽しく読ませていただきました。ところで、山本寛氏と祝一平氏は同一人物ではないのでしょうか。

上田　考一(22)福岡県

とりあえずアヤシイ関係、とだけいっておきましょう。

◆「X-OVER NIGHT」の高原さん。あんまり○立，日○と書かないでください。どうせ書くならバレーボールに注いでいるお金を社員の給料に回してあげなさい，とでも書いてくださいよ。毎月70～80時間残業してもたいした金額をもらえないのですから。給料が安くて休みがない，そのうえ作業もきつく，ついでにロボットにもて遊ばれるのに，ボーナスもあまりよくない。○立でいちばん人出不足，かつ，いちばん大変な工場の現場の人より。　伊藤　徹(22)千葉県

忙しいのはいずこも同じ，ってところですね。

◆台風の季節が近づいてきました。今年は去年のようなことが起こらないことを祈りたいです。それは台風が上陸したときに家で飼っている犬が，キャンキャン鳴いて窓をガリガリこすり、まるで「入れてくれ～！」といっているようなことがあったのです。かわいそうでしたが，少しだけ面白かった。そのころホンニャアはなにをしていたんでしょうか。聞いてみたいものですね。　小林　貴秀(15)新潟県

できるなら，今年は優しく犬君を家に入れてあげましょうよ。

◆今年の夏もまた，異常気象の気配が濃くなってきましたね。ところで最近知ったんですけど，夏バテって夏にバテることじゃなくて，初秋に入り夏場にたまった疲れが一気にくることらしいです。読者の皆さん，あまりパソコンをいじってばかりいないで外でちゃんと遊びましょう。編集部の皆さん，お仕事休むことも大切ですよ。もちろんOh!Xの中身が薄くなっていいというわけじゃないけど。　岡部　誠(27)福井県

もちろん，毎月楽しみにしてくれている読者のためにがんばってますからね。

◆最近変な夢を見た。パソコンショップに見慣れないマシンが置いてあるのだ。なんとそのマシンは，X1turbo＋CD-ROMというX1twinのようなマシン。シャープ恐るべし，と思ったが夢なのには変わりなかった。角谷　光憲(18)愛知県

X1turboにPC-Engine＋ROM²を瞬間接着剤でくっつければ，夢のマシンが……できるわけないか，残念。

◆アンケートハガキの下にある所有機種欄にMIDI楽器が新しく加わりましたね。MIDI楽器を3台持っている私としては嬉しい限りです。また，私は趣味で作曲をしています。特に映像と音楽の同期には興味深いものがあり，映画サークルに入っている友人の作品の音楽を作ったりもしました。その私が思うのですが「Creative Computer Music入門」は面白くありません。それに初心者が読んでもあまり役に立たないと思います。なぜなら変に難しいからです。作曲は一部の人だけの能力だけでなく，誰でもすぐにできることなのですから。梅津　信幸(20)京都府

理論による裏づけも大切でしょう。

◆最近私は「涙腺」がもろい。「レミングス」の2面の曲を聴くたびに「出たな!!　ツインビー」をやるたびに涙がポロポロ出る。一時的なことならよいけど，この先ますますひどくなって1日中ポロポロなんてことになったらどうしよう。などと考えている今日この頃でした。

猪原　裕太(23)東京都

涙を流すだけでは芸がありません。いなかっぺ大将の大ちゃんみたいに，涙でアメリカンクラッカーをやることに挑戦だ。

◆私は絵が描けない。字も汚くて下手だ。いったいどうやってOh!Xに投稿すればいいのだろうか。初めてパソコンをやってから，まだ3カ月の私にプログラムができるわけもないし。なにかよいアイデアはありませんか。

石川　勝敏(16)北海道

こうやって投稿すればいいんですよ。

◆本日（6/17）スーパーウォークマンの音を聴いてきました。感想は，す・ご・いです。現時点では40分テープしかないそうですが，もうすぐ120分テープができるそうです。本体は思ったより大きいですが，約10万円という価格（普及すればもっと安くなるのに）はすごいです。でも，小さすぎてなくしそう。

加藤　恵吾(16)愛知県

確かに，持ち運びには非常に便利だけどテープ管理は大変そうだなあ。

◆いままで私の髪型は俗にいう“オタクヘア”だった。前髪が鼻の頭にまでたれ下がり，1回散髪に行けば2カ月は持つような髪だった。しかし，先日部活の後輩に「先輩，急に老けましたね」といわれた。ちょうど髭を3日ほど剃っていなかったときなのでそう思ったらしい。男にそういわれても別になんとも思わないが，さすがに女の子にまでいわれるとなるとちょっとショックを受ける。さっそく次の日刈り上げてもらった私は若々しく……なったのかなあ。

幸　俊威(17)大阪府

若々しい高校生。すごく当たり前のような気がするんですけど。

◆奥さんにつわりが始まり，それに伴って家事が増え，パソコンに触れる時間がドンドン減ってきました。悲しい限りです。でも，食器を片づけながら子供の名前をあれこれ考えていると，ひょっとして俺は幸せものなのかなあ，と思ったりします。ところでなにかいい名前ありませんか。

谷本　俊康(25)東京都

隣にいてこんな話をされたら，「この，幸せもの」とかいいながらヘッドロックをかけてしまいそう。

◆大感激！　6月19日妻の出産に立ち会いました。脂汗の1時間，産声のあと自分の疲れもかえりみず，か細い声で「五体満足ですか？」と聞く妻に思わず涙が出てしまいました。そういうわけで，生活のパソコンのプライオリティは下がる一方です。しばらくはX68000を遠くに置いて眺めていようと思ってます。でも，それは少しも悲しいことではなく嬉しいことなのです。最後にひと言，6月19日午前11時43分，長女誕生3560g，命名「知恵」。とってもかわいいです。

石川　仲宰(33)福岡県

こちらも幸せいっぱいのハガキ。お子さんと一緒に楽しいパソコンライフを送れるといいですね。

◆私は受験生であった。大学にも入って受験は人ごとのように思っていたのに，まだ大学院の受験がありました。高校，大学の入試よりも早く9月に行われます。でも，受験雑誌Oh!Xを読んでいれば大丈夫ですよね。

松永　貴輝(21)大阪府

受験雑誌Oh!X，誰がいいだしたのかわかりませんが，松永さんうかうかしているとどうなっても知りませんよ〜。

◆7月号99ページの上にある広告の女性の持っているものを見て，はっと気づいた。QDとはクイックディスクのことなんですね。私が中学生のときにあった，ファミコンのディスクシステムに使われているものと同じ円盤とは，懐かしいなあ。ところで，同ページの上の女性，下の小さな広告の男性，できれば女性のオマケみたいなおじさん。それぞれの名前がわかる人はご一報を。

志賀　宗一(18)愛知県

上の女性は倉沢敦美ちゃん，下の男性は宍戸錠，おまけのおじさんは，その昔「パソコンサンデー」という番組に出ていた，通称ドクターパソコン宮永好道さんです。

◆友人がXCを買ったのでプログラマーズマニュアルを見せてもらいに友人宅へ行くことが多くなった。そのためいつも帰りが遅くなってしまうので，祖父が「パソコンを使うのに必要なんだったら，買ってやるから早めに帰宅しろ」といってポンとXCを買ってくれた。思いがけないことだったので非常に嬉しかった。買って帰る途中，あまりの重さに右手がつってしまったけど，なんとも嬉しい痛みだった。

千葉　浩貴(19)宮城県

自分の好きなことを認めてくれたおじいさんのためにも，がんばってプログラミングに励みましょう。

◆先日，電力実験というものをやりました。それはとても怖い実験で過去に死者も出ています。最大で2kWぐらいのパワーを出しました。自分の班は結構無事に実験が終わり，死者，負傷者ゼロでした。でも，1回だけ大型の可変抵抗器をいっぱいに回してショートさせてしまい，その端子を半分溶かしてしまいました。あ〜，死ぬかと思った。

廿日出　悟(20)東京都

やっぱり大学って変なところなんですね。

◆私には気になっていることがあります。それは風もないのに障子や戸が，激しい音を出すことがあるのです。去年夜中に突然このことが起きたときには本当に驚かされましたが，今年は昼間たまに起きているだけです。原因不明ですが，私は勝手に「我が家にだけ地震」と思っていたりするのです。

藤原　彰人(22)岡山県

それはただのラップ音かもしれません。ま，ちょっと低級霊が遊んでいる程度ですから気にしなくていいでしょう。

◆アンケートハガキにある「編集部行」は「編集部御中」に直すのはわかりますが，あの「(受取人)」はどうすればいいのでしょうか。僕はいつも消してしまうんですけど，正しい方法を教えてください。

丸山　淳平(18)神奈川県

郵便局が受取人を明確にするためのものですから，消さなくても結構ですよ。

◆突然，荻窪氏が顔写真を公表されましたね。Oh!Xだけでなく，某雑誌の特集記事（こちらはカラーで特大判）にも掲載されていました。名を惜しむこと，貧者が珠を持つが如くの荻窪氏が，なぜ大衆の前に姿を現したのか。

1）　すべては荻窪氏流の冗談である

懐が深い荻窪氏のこと，読者の反響を観察し絶妙のタイミングで「あれは影武者だ」と真実を告げる。これによって「大人のためのX68000」はネタに困ることがなくなる。

2）　近い将来に結婚を予定している

まったくの直感です。

なんてね，以上大変失礼しました。

横川　和美(28)埼玉県

3）　単なる出たがりや。という可能性もなきにしもあらずですよ。

◆7月号で私の自画像1号（顔が若い！）を載せていただき，ありがとうございます。あれからもう6年……懐かしく昔を振り返らせていただきました。これからも「ちゃだワ」への自画像攻撃を続けさせてもらいますので，よろしくお願いします。また，5年ぶりぐらいにイラストの投稿でもさせてもらいます（し，仕事が〜）。

酒井　強(24)三重県

そのうち教師生活十数年，Oh!X投稿歴も十数年になったりして。

◆7月号の「SHIFT BREAK」のA.T.さんへ。ありますよ，そういうプリンタ。チノンの「CTK-120」です。高速，軽い，バッテリー駆動，といたれりつくせりの300dpiで178,000円だそうです。

大上　幸宏(19)鹿児島県

◆7月号の「SHIFT BREAK」のA.T.さんへ。PC-9801などに使えるプリンタでチノンから熱転写ページプリンタ（型番は忘れた）が出ています。専用インクリボンでA4サイズ幅を一気に印字できるヘッドを持っているそうです。でも1枚当たりのコストが100円を超えるという，超高い

▲小川　裕美　山口県
身についた習性，イラスト常連の性でしょう。それにしても，小川さんのイラストはちょっとずつタッチが変わっていて面白いなあ。

▲肥後　有紀子　大阪府
あまりのインパクトの強さに採用してしまった，実録!?　姉妹4コマ。あんまりケンカを売らずに仲よくOh!Xを読んでくださいね。

コストです。　　　　　髙橋　正博(28)栃木県

　カラーコピー並のコストを持つ熱転写プリンタとはちょっと驚きですね。

◆ど，どうも。7月号で久しぶりにイラストが載った住友です。過去数回載ったことがありますけれど，ここまで「恥ずかしい」と思ったのは初めてです。友人たちにもバッチリうけました。それにしても，う～，すごいよお。自分が描いたものとは思えん。でもうけているからいいか。でもすごい。岩瀬さん，小川さん，あなた方は，どうかこんなふうに道を踏み外さないでくださいね（なんだかなあ）。

住友　智代(19)愛媛県

　道を踏み外すというと，やっぱりJU○Eとかさ○とかの世界ですか。う～ん，すみやかに社会復帰することを祈っています（そこまでいってないって）。

◆なにがおかしくなったのか，何度OS-9をインストールし直してもトラブるようになってしまった。バージョンアップすればなんとかなるのだろうか。それとも修理だろうか。ああ，どんどん工程が遅れていく～！　う～んどうしよう。よし，ここはHuman68kでLINK前までなんとか進めて……おや，皆さん会社員ですか？　いいですねえ（うるうる）。

只野　はるみ(26)神奈川県

　そういうあなたはフリーのプログラマさん。でもフリーという言葉にちょっとだけ憧れを感じないこともないんですよ（たとえつらいとわかっていても）。

◆酔っ払って焼き鳥食いながら，ハゲた頭に花束をガムテープでくくりつけ，道ゆく人々に静かに語りかけていた面白いおじさんをこの前見かけました。　　　太田　敬三(22)東京都

　そのうち街の名物おじさんとなるんでしょうか。世の中いろんな人がいますねえ。

◀井上　綾子　東京都
モデル化のうまさに感心してしまうSIMシリーズ。今回は残念な結果に終わってしまったようですが，また新しい文明を目指してがんばりましょう。

ぼくらの掲示板

●掲載ご希望の方は，官製ハガキに項目(売る・買う・氏名・年齢・連絡方法……)を明記してお申し込みください。
●ソフトの売買，交換については，いっさい掲載できません。
●取り引きについては当編集部では責任を負いかねます。
●応募者多数の場合，掲載できない場合もあります。
●紹介を希望されるサークルは必ず会誌の見本を送ってください。

仲間

★この度，X68000ディスクメディアサークル「BBQ」を発足するにあたり会員を募集します。初心者大歓迎です。内容はなんでもありの自由な表現の場，といった感じにしたいと考えています。ただ，フリーウェアを配布するだけが目的ではありません。興味のある方は，5インチ2HDディスク2枚と返信用封筒を同封のうえ，下記の住所までお送りください。〒305　茨城県つくば市春日4-13-30　明峰ハイツA-111　野口　友則(20)

売ります

★X68000用ハンディスキャナ「X68 HGS-68」を18,000円で売ります。マニュアル，ソフト，付属品，箱すべてあり。新品同様です。連絡は往復ハガキでお願いします。〒192-03　東京都八王子市大塚58-69　安保　徹(22)

★X1用FDDインタフェイス（「CZ-503F」に付属していたもの）を送料別1,000円程度で売ります。連絡は官製ハガキでお願いします。〒355　埼玉県東松山市和泉町4-35　鈴木荘202号　池田達也(22)

★300bpsモデム「CZ-8TM1」を4,000円，1,200bpsモデム「PV-A1200MK3」（アイワ）を6,000円，ともに送料込みの値段で売ります。マニュアル，付属品，箱すべてあり。完動品です。連絡は往復ハガキでお願いします。〒245　神奈川県横浜市戸塚区深谷町25　深谷団地10-55　黒武者健一(22)

★アイテック「ITX-680」（40＋40Mバイト）を送料込み46,000円で売ります。外観上，付属品，箱あり。早い者勝ちです。また，自作の314MバイトSCSIハードディスクドライブを190,000円以上で売ります。こちらは先着3名のうち，最も高い方に売ります。本体は手製で黒のアクリル板(5mm厚)，ドライブはシークタイム12ms，データ転送速度4Mバイト/sのOEM新品を使用しています。外形寸法は，12cm（幅）×14cm（高さ）×25cm（奥行）でSCSIのID番号スイッチも装備しています。SCSIケーブル，ターミネータ付き，SCSIコネクタは入出力を用意しています。購入後のサポートも受けつけますし，1週間以内でしたら返品も受けつけます。〒361　埼玉県行田市持田5-13-2　吉本　邦雄(44)

★Roland「U-220」を47,000円以上で，Roland「CM-32L」を35,000円以上で売ります。いずれも，トラブル防止のため手渡し希望します。「U-220」は箱なし，そのほかすべてありで，「CM-32L」は箱そのほかすべてあります。高く買ってくれる方優先です。まずは往復ハガキで連絡をお願いします。〒331　埼玉県大宮市大成町2-319-2　野崎　徹

買います

★X1用FM音源ボード「CZ-8BS1」と，RS-232Cマウスボード「CZ-8BM2」をそれぞれ10,000円以内で買います。また，カラーイメージボード2「CZ-8BV2」を17,000円以内で買います。いずれも送料込みにしてください。連絡は官製ハガキでお願いします。〒355　埼玉県東松山市和泉町4-35　鈴木荘202号　池田　達也(22)

★X68000用MIDIボード＋RolandMIDI音源モジュール「CM-32L」を43,000円以内（送料込み）で買います。付属品付きで完動品であれば，多少の汚れ，傷，箱なしは可。連絡は官製ハガキでお願いします。〒761-04　香川県高松市三谷町884-4　西池　陽一(14)

★X1用立体映像セット「CZ-8BR1」を送料込み9,000円程度で買います。付属品付きで完動品であれば，傷，汚れ，箱なしでもかまいません。連絡は往復ハガキでお願いします。〒980　宮城県仙台市青葉区子平町18-17　YAコーポ103号室　小松　健一郎(18)

★X68000用拡張I/Oボックス「CZ-6EB1-BK」を送料込み40,000円くらいで買います。まずは，箱，説明書，傷の有無と希望価格を書いて，往復ハガキで連絡してください。〒762　香川県坂出市白金町3-8-34　佐藤　崇司(18)

★X68000版「スペースハリアー」「ドラゴンスピリット」の広告用特大ポスターを送料込み各8,000円で買います。折り曲げてないもの，傷，汚れのないものに限ります。連絡は往復ハガキでお願いします。〒408-03　山梨県北巨摩郡武川村上三吹345　輿石　学(21)

バックナンバー

★Oh!X1990年10月号，1991年3月号を各1,000円（送料込み）で買います。切り抜きのあるものは不可。連絡は往復ハガキでお願いします。〒930　富山県富山市新園町336　新園ハイツ405号　杉林　隆志(17)

編集室から from E・D・I・T・O・R

DRIVE ON

このコーナーでは，本誌年間モニタの方々の意見を紹介しています。今月は７月号の内容に関するレポートです。第８期モニタの皆さんによる最初のレポートを紹介します。

●９月号から始まるCGA講座ですが，せっかくOh!Xの読者全員にDōGA CGAシステムがいき渡ったので，今度の連載はいちから始めてほしいです。実際，私は以前の連載のときにはCGAシステムを持っていないこともあり，ほとんど読んでいませんでした。たまに目を通しても何がなんだかわからないものですから，余計に離れてしまいます。そして，CGAという分野はひとりでやろうとすると，ものすごく多才でなければなりません。単に絵が描けるだけではだめで，全体の構成とかコマ割，動きの表現，へたをすると音楽とのシンクロまでやらなくてはならないでしょう。DōGAのようにチームを組んでいればともかく，最初はひとりで始めるのが普通です。１つひとつを丁寧に教えてほしいものですね。

中村　健(22)X68000 ACE-HD，PC-386GS，MSX2＋　埼玉県

●７月号から新しく「ハードウェア工作」が始まりました。このテの連載で，まず最初につまずくところといえば，理論や考え方以前のレベル，つまり“単語”ではないかと思います。実際に自分自身の昔のことを（いまでもそうだけど）考えてみると，見たこともないような単語が出てきた時点で思考の流れが止まってしまう傾向があるのです。囲み記事でもいいですから，なんらかの形でフォローしてください。また，具体的なテーマとして個人的には，TTLを組み合わせてCPUを作り上げるのがベストではないか，と考えています。ゼロから組み上げればかなり実感が湧くと思うのです。

中島　奨(25)X68000 PROII，MZ-1500，Macintosh SE/30　北海道

●実録TORNADO秘話「4DCGへの招待」はとても面白かったです。１分45秒の映像にかける意気込みが，伝わってきて７ページをあっという間に読んでしまいました。「何をしたいのか」「最後に何を得たのか」という過程もきっちり書いてあり，カットや写真，注釈も効果的で読み手をグイグイ引きつけてくれます。文月さんという人はとても素晴らしい人なんでしょうね。次回作を期待しています。

村上　晃(23)X68000 XVI　岡山県

●「ついに姿を見せたV70ボード」を読んで，アセンブラやＣコンパイラなど開発環境がしっかりしているようなので，なかなか面白そうだと思いました。レイトレーシングみたいに膨大な計算をする人には，強い味方になるでしょう。実際にプログラムリストが載っているのを見て急に親しみが出てきたし，ほしいと思った人も結構いたかもしれません。しかし，値段を見てため息をついた人も同じくらいいたことでしょう。ほかに比較すべきものを知らないので，V70ボード248,000円，開発キット68,000円という価格が高いか安いかよくわかりませんが，個人でポンと出せる金額ではないことは確かです。

矢野　啓介(19)MZ-2500　北海道

ごめんなさいのコーナー

５月号　（で）のショートプロぱーてぃ
(michelle plue)

1790，1830，1980行のグラフィックキャラクタが正しく印字されていませんでした。リスト１のように訂正してください。

８月号　よいこのSX-WINDOW

リスト５の一部が不鮮明でした。正しくはリスト２のとおりです。

●Z-MUSIC ver.1.10

強制同期コマンドの動作不安定，モジュレーションディレイ誤差の訂正です。以下のアドレスをMAC.Xで修正してください。

1CCA:34→60	7A7E:66→60
1CCB:01→00	7AD8:66→60
1CCC:48→29	9AD2:13→43
1CCD:42→FA	9AD3:40→FA
76A6:66→60	9AD4:FF→D2
773A:66→60	9AD5:48→42
7798:66→60	9AD6:43→13
7876:66→60	9AD7:FA→40
79D4:66→60	9AD8:D2→FF
7A28:66→60	9AD9:3E→48

リスト１

```
1790 FOR I=0 TO 38 STEP 2:LOCATE I,1:PRINT A$(19);:LOCATE I,23:P
RINT "▚□";:NEXT
1830 LOCATE 0,21:PRINT"□"+CH2$+"〒";:LOCATE 39,21:PRINT"▚"+CH2$+"
生";
1980 DATA ﾀﾁ,ﾐﾑ,ﾂﾃ,ﾒﾓ,ﾄﾅ,ﾔﾕ,ﾆﾇ,ﾖﾗ,ﾈﾉ,ﾘﾙ,ﾊﾋ,ﾚﾛ,ﾌﾍ,ﾜﾝ,ﾎﾏ,ﾞﾟ,●○,▒土,
♠♥,金木,◆♣,水火,▲,月日,╳▀,時分,ｹｹ,ｹｹ,▄ ▀,秒年,▚□,生〒, ▄▀,円人,ｽｼ
```

リスト２

```
      bm0=GMGetBitmap();        /* 元のビットマップ（テキストタイプ） */

      bm.bmKind         =G_GRP; /* グラフィックタイプのビットマップを作る */
      bm.bmRect         =bm0->bmRect;
      bm.base           =0xc00000;
      bm.line           =2048;
      bm.opt.bRatio     =0x8000;

      setPalet();               /* パレットを設定 */

      GMSetBitmap( &bm );       /* グラフィックタイプのビットマップに変更 */

      /* ここからはグラフィック画面に描画される */

      p.x_y=0x00100010;
      PutImg(G_GRP,(rectImg*)neko_gr,p);
      p.x_y=0x00100040;
      PutImg(G_GR2,(rectImg*)neko_gr2,p);
      p.x_y=0x00100070;
      PutImg(G_GR3,(rectImg*)neko_gr3,p);
      p.x_y=0x001000a0;
      PutImg(G_TXT,(rectImg*)neko2,p);

      GMExgBitmap( bm0 );     /* テキストタイプのビットマップと交換 */
```

バグに関するお問い合わせは
☎03(5488)1311(直通)
月～金曜日16：00～18：00

お問い合わせは原則として，本誌のバグ情報のみに限らせていただきます。入力法，操作法などはマニュアルをよくお読みください。

また，よくアドベンチャーゲームの解答を求めるお電話をいただきますが，本誌ではいっさいお答えできません。ご了承ください。

豪華な皿には豪華な料理がほしいよね

▶今月号の特集では，数値演算の高速化を中心にいろいろなアプローチをしてみました。これらの記事を読むとおわかりになるでしょうが，コンピュータの性能はハードウェア×ソフトウェアで成り立っている，ということです。高性能のハードウェアに高機能のソフトウェアがあれば「鬼に金棒」，向かうところ敵なし。皆さんも自分のマシンを見つめ直し，足りないものを補ってどんどん活用してください。

▶今月から予告どおり「DōGA CGアニメーション講座 ver.2.50」が始まりました。そろそろ，申し込んだDōGA CGAシステムのマニュアルが届いていることもあり，連載開始を待たずに活用している人もいるかもしれませんね。それでも初めて触る人にとっては，わからないことだらけでしょう。そんな人は，この連載と一緒に基礎からこつこつと学んでいってください。そして，それぞれ自分の中に芽生えたイメージを具現化するため，CGAシステムを使ってみませんか。

▶さて，まだまだ暑い日が続きますが，皆さんお元気でしょうか。さっそく暑さにやられた，なんてことはないでしょうね。

そして，夏にあったさまざまな体験，面白い出来事などありましたら，どんどんアンケートハガキに書いてください。もちろんプログラムの投稿も待ってます。学生の皆さんはせっかくの夏休み。時間を有効に使って何かプログラミングしてみましょう。目標は大きく，志は高くもってプログラミングできる絶好のチャンスですから。

▶それから，暑中見舞いのハガキをくださった皆さん，どうもありがとうございます。なかには6月から送ってきた気の早い人もいますが，夏らしいきらびやかなイラストがたくさん送られてきて，編集部一同喜んでいます。とりあえず，10月号で予定されている「Oh!X reader'sぎゃらりぃ 暑中見舞い編」で紹介するつもりです。

それでは，来月号をお楽しみに。

投稿応募要領

- 原稿には，住所・氏名・年齢・職業・連絡先電話番号・機種・使用言語・必要な周辺機器・マイコン歴を明記してください。
- プログラムを投稿される方は，詳しい内容の説明，利用法，できればフローチャート，変数表，メモリマップ（マシン語の場合）に，参考文献を明記し，プログラムをセーブしたテープ（ディスケット）を添えてお送りください。また，掲載にあたっては，編集上の都合により加筆修正させていただくことがありますのでご了承ください。
- ハードの製作などを投稿される方は，詳しい内容の説明のほかに回路図，部品表，できれば実体配線図も添えてください。編集室で検討のうえ，製作したハードが必要な場合はご連絡いたします。
- 投稿者のモラルとして，他誌との二重投稿，他機種用プログラムを単に移植したものは固くお断りいたします。

あて先

〒108 東京都港区高輪2-19-13 NS高輪ビル
ソフトバンク出版部
Oh!X「㋢㋣㋮㊔」係

S H I F T · B R E A K

▶理系の大学生で学校の研究室に所属している。そう自己紹介をすると必ず「どんな研究ですか」と質問されるのだが，これがまったく始末におえない。普通の人にはわからないだろうと，苦心して平易に話すのだが，100人中127人くらいの人が「ふ〜ん」とかいって理解しない。わからないんだったら最初から聞かないのが礼儀だと思うぞ，うんうん。(八)

▶ラーメン屋を教えてくれた皆さん，ありがとう。いつか食べに行こうと思っています。最近になって時間に拘束されることがなくなったので，ソースの汚いSV.Xをリライト＆バージョンアップしています。それとMAGICもオイラー角に対応させなきゃいけないし……。やらなきゃいけないことはたくさんある。また忙しくなりそう。(H.K.)

▶MIYA-NETというところでZMUSICのサポートをしてくれています。私も全面的に協力していまして最新版のZMUSICやZMUSIC支援ツールや演奏データなどがアップロードされています。GUESTでも利用可能ですので遊びにきてください。電話番号048-648-9801（入会無料，回線数4，24時間ただしGUESTはPM9：00〜AM1：00は利用できません）(善)

▶おっと，いきなりカミキリ虫が部屋の中に飛び込んできやがった。この前は家の近所のゴルフ練習場でカブト虫を捕まえたし。ああ，なんて素晴らしい環境のもとで暮らしているんだ(本音)。東京に限らず，自然から遠ざかってしまった人間が多いのに，私が住んでいる街はまだまだ大丈夫のようだ。いまでもザリガニはとれるのだろうか。(S.K.)

▶レイトレ用に編集室からCompactを借り出した。持ち帰れるお手軽サイズ。PROIIからRAMを2Mバイト移植した。フロッピーベースじゃしようがないので，いまさらながらSyQuestした。ソフトのインストールがまた大変。PROIIにはSCSIボードを付けていない。手作業で3.5インチフロッピーに1枚1枚メディアコンバートってのは情けなかった。(A.T.)

▶いやあ，朝日新聞は相変わらず面白い。ほんと。どーすんでしょ，あんなに笑える記事ばかり載せちゃって。新しいパソコンとか，マルチメディアとか，そういうものに力を入れるってのは，無理やり1面を飾ることじゃなくて，コンピュータ業界に強い記者を育てるってことじゃないのかねえ。X68000のウイルス事件以来進歩ないぞ。(K)

▶いままで少女マンガを買うことに抵抗はなかったが，セーラームーンの単行本を買うときは少し恥ずかしかった。結局，お酒に酔った勢いでしか買うことができなかった。読んでみてびっくり，三人のセーラー服戦士のほかにもセーラージュピターが登場していた。これから先，ビーナスとかサターンとかも登場するのだろうか。展開が気になる。(KO)

▶最近，社会人になった友人に会うと，ほとんど「仕事がつらいなあ」という言葉が出てくる。学生時代とのギャップもあるだろうが，自分の思ったことができないという状況にがまんできないようだ。そんななかで「そうだよなあ，やっぱつらいよなあ」といいながら，やりたいことができる職場にいる僕は幸せなんだろうな。(J)

▶文月さんが家にやってきたので，AMIGAとMacintoshを動かしてみせてあげたら，マウスを畳の上で転がしている姿にひたすら爆笑されてしまった。寺尾さんにそのことを話すと，「和風でいいじゃないですか」と少しヘンなフォローをしてくれた。でも，やっぱり机を買おうかなあ。マウスボールにイグサがささってしまうし，っていうのはウソ。(A)

▶ああ，今年は夏が短いっていうのに，まだ海にもプールにも行ってないよぉ。あ，そっか，あたし会社辞めるんだっけ。なんだ，じゃ，これから毎日行けるじゃん。さぁて，夏を満喫したあとは，車とバイクの免許でも取りに行って，それからニューヨークに行って……。なんて，金がそんなにあるわきゃないって。それでは皆さん，さようなら〜。(E.O.)

▶と，いうわけでE.O.嬢は今号で引退。いろんな職をこなしてきた彼女だから立派に兼業主婦をするのでしょう（しかしアンナミラーズに勤めてたというのはインパクトが大きかった）。とりあえずお疲れさま。え，なにこれ？ ちゃんと新卒から6年って書いたよね？ そうか，世間ではもうお父さん扱いされる歳なのか……。(メガネはかけていないU)

▶486/33のAT互換機でWindowsを動かす。最も速いといわれるWordの高速モードでただのテキストスクロールを計ったら，X68000XVIでSX-WINDOWのエディタ.Xを使った場合とほぼ同じ速度であった。条件は多少違うものの，あらためてSX-WINDOWの軽さを実感。もっとも私はエディタしか使わないからそんな能天気なこといってるんですけど。(T)

microOdyssey

なんだかここ1年ほど，妙に忙しいと感じていた。もちろん仕事はそれなりに忙しかったし，仕事以外の用事も多かった。でも，学生の頃は1日に7～8件の用事をこなしていても忙しいとは感じてなかったと思う。睡眠時間が3時間前後だったにもかかわらず，だ。それから比べれば，いまの生活なんて遙かにラクに感じるはずなのに，逆に忙しいと感じていたのだ。う～ん，仕事が不規則なせいかなあ，それとも私生活がヘンにゴタゴタしていたからかなあ，などと要因をしばらく考えていたのだけど，この前はたと気がついた。

学生の頃といまとでは，時間の流れ方が違うのだ。充実度の度合といったほうがいいのかもしれないな。大好きなことを無我夢中でやっているときは時間がすぐにたってしまうのに比べて，そうでないときの時間の流れ方はやたらと遅く感じてしまう，そういうのってなんとなくわかるでしょ？　実際，大好きなことよりも気合が入っていないぶんだけ，会社にいる時間はやたらと長いし，不透明な時間が多くあるように思えてきたのだ。だからといって，あたしはこの編集という仕事がけっして嫌いなわけじゃあない。むしろ好きだと感じている。でも，編集の仕事よりもやっぱり音楽が好きだ！　ということがはっきりとわかってしまったのだ。

とはいえ，やっぱり仕事はしなくちゃだから，ここは頑張って両方に100%の情熱を注ごう，と努力してみた。でも，そう努力すればするほど，音楽にしても仕事にしても中途半端であいまいに流しているような感じに襲われ，絶望感と劣等感だけがまとわりつくようになっていった。

ほんとはあいまいでもよかったのかもしれない。だいたい，いまの世の中なんてあいまいなことだらけなんだもの。でも，あたしはけっしてあいまいにすることがいいこととは思えなかったし，もちろんいまもそう思っている。

そういえば，1週間ほど前に恋人と話しているときに，彼が「まあ，いいや」とあいまいな言葉を口にしたことがとてもショックだった。彼は何の気なしにいったのだろうけど，その瞬間に信じ合っているはずの2人の関係ですら，とても希薄なものに感じてしまったのだ。でも，彼が悪いわけではない。この混沌の時代にあいまいでいられないあたしのほうが悪いのだろう。いくら時代遅れなアナログ野郎といわれても，あたしは「まあ，いいや」と流れに身をゆだねて生きるより，自分の意志でちゃんと歩いていきたいと思うから。強情っ張りなんだろな，あたしってば。

結局あたしは音楽1本に絞ることに決めた。やっと答えが得られたねって感じ。ずいぶんと遠回りをしてしまったなあ，と思う。でも，たしかにいまのあたしは音楽がやれるだけで幸せだし，ほかには何もいらないから。こんな気持ちでこの編集部にいることも申し訳ないと思うしね。でもって，ちょうど音楽に浸れるチャンスを与えてくれる人がいた。その代償として，どうやら自分の姓名を変えることになりそうだけど，ま，いいか。

ふと思い出したけど，昔付き合っていた男が別れる最後の最後に「君は音楽を捨てられないよ」といっていたな。はは，そのとおりになってしまったなぁ。 (E.O.)

1992年10月号 9月18日(金)発売

特集　DTMへの招待
・MIDI音源を使いこなす
・Z-MUSICの活用
拡張版Oh!X LIVE in '92
試用レポート　X68000用CD-ROMドライブ
全機種共通システム
Small-C用横スクロールシューティングゲーム　他

バックナンバー常備店

東京	神保町	三省堂神田本店5F	03(3233)3312
	〃	書泉ブックマートB1	03(3294)0011
	〃	書泉グランデ5F	03(3295)0011
	秋葉原	T-ZONE 7Fブックゾーン	03(3257)2660
	八重洲	八重洲ブックセンター3F	03(3281)1811
	新宿	紀伊国屋書店本店	03(3354)0131
	高田馬場	未来堂書店	03(3209)0656
	渋谷	大盛堂書店	03(3463)0511
	池袋	リブロ池袋店	03(3981)0111
	〃	西武百貨店9F コンピュータ・フォーラム	03(3981)0111
神奈川	横浜	有隣堂横浜駅西口店	045(311)6265
	〃	有隣堂ルミネ店	045(453)0811
	藤沢	有隣堂藤沢店	0466(26)1411
神奈川	厚木	有隣堂厚木店	0462(23)4111
	平塚	文教堂四の宮店	0463(54)2880
千葉	柏	新星堂カルチェ5	0471(64)8551
	船橋	リブロ船橋店	0474(25)0111
	〃	芳林堂書店津田沼店	0474(78)3737
	千葉	多田屋千葉セントラルプラザ店	0472(24)1333
埼玉	川越	黒田書店	0492(25)3138
	川口	岩渕書店	0482(52)2190
茨城	水戸	川又書店駅前店	0292(31)0102
大阪	北区	旭屋書店本店	06(313)1191
	都島区	駸々堂京橋店	06(353)2413
京都	中京区	オーム社書店	075(221)0280
愛知	名古屋	三省堂名古屋店	052(562)0077
	〃	パソコンΣ上前津店	052(251)8334
	刈谷	三洋堂書店刈谷店	0566(24)1134
長野	飯田	平安堂飯田店	0265(24)4545
北海道	室蘭	室蘭工業大学生協	0143(44)6060

定期購読のお知らせ

Oh!Xの定期購読をご希望の方は綴じ込みの振替用紙の「申込書」欄にある 新規 継続』のいずれかに をつけ，必要事項を明記のうえ，郵便局で購読料をお振り込みください。その際渡される半券は領収書になっていますので，大切に保管してください なお，すでに定期購読をご利用の方には期限終了の少し前にご通知いたします。継続希望の方は，上記と同じ要領でお申し込みください。

海外送付ご希望の方へ

本誌の海外発送代理店，日本IPS(株)にお申し込みください。なお，購読料金は郵送方法，地域によって異なりますので，下記宛必ずお問い合わせください。

日本IPS株式会社
〒101 東京都千代田区飯田橋3-11-6
☎03(3238)0700

Oh!X 9月号

■1992年9月1日発行　定価600円(本体583円)
■発行人　孫　正義
■編集人　橋本五郎
■発売元　ソフトバンク株式会社
■出版事業部　〒108 東京都港区高輪2-19-13　NS高輪ビル
Oh!X編集部　☎03(5488)1309
出版営業部　☎03(5488)1360 FAX 03(5488)1364
広告営業部　☎03(5488)1365
■印　刷　凸版印刷株式会社

©1992 SOFTBANK CORP. 雑誌 02179-9 本誌からの無断転載を禁じます。
落丁・乱丁の場合はお取り替えいたします。

SOFT BANK

ソフトバンクの
新刊

ソフトバンク
出版事業部
東京都港区高輪2-19-13
TEL03(5488)1360

Cプログラマのためのｃ++入門

柴田望洋 著
定価2,900円

「C:98スーパーライブラリ」の柴田望洋氏が書き下ろした、最も簡潔で正確なC++の解説書。CからC++へ、その拡張の必然性とC++の特長を、体系的に解説。

Btrieveシステムガイド

MML&VAINS 著
定価4,800円

NetWare上で標準装備されるデータベース、Btrieveについて、その高速度を誇る全貌を解説し、dBASEで作成したシステムをLANに移行するための解決策を提示。ディスク付。

Macintoshユーティリティとつきあう本

狩野都 著
定価1,600円

System7の登場で操作環境を大きく変えつつあるMacintoshについて、新しい環境下でのユーティリティの利用法をユーザーのニーズに応じて目的別に解説。

Practical C Programming
現実的なCプログラミング

Steve Oualline 著
岩谷宏 訳
定価3,600円

米国オーライリー社のNutshellシリーズの一冊。初級から中級を対象に、プログラムの仕様決定から改版まで、Cプログラミングの全サイクルを明快に解説した定本的入門書。

パソコンLANでどこまでできるか

杉山育央 著
定価1,800円

パソコンLANで実際に何ができるのかを徹底的に回答。導入や運営についてのチェックポイントや実例をあげながら、現在のパソコンLANの実力を余すところなく語る。

NetWare API入門

荒井文吉 著
定価3,600円

NetWareプログラミングの実践的入門書。サンプルプログラムの作成を通し、NetWareのサービスを利用するアプリケーションの作成方法を具体的に解説。5"ディスク付。

MS-DOS Ver.5 環境整備術
フリーソフトウェアでつくるDOS-シェル環境

アルシーブ 著
定価2,400円

MS-DOS Ver.5とアプリケーションだけでシステムを構築しているユーザーを対象に、より快適なファイル／ディスク処理の環境を提供するソフト付き書籍。

MS-DOS実用マスターシリーズ④
新MS-DOS Ver.5.0入門　ビギナー編

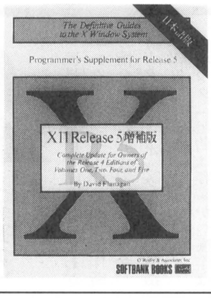
林晴比古 著
定価1,900円

好評シリーズをMS-DOS Ver.5に合わせて全面改訂し、初めてパソコンに触れる人を対象に基本的な知識を完全解説。実務に不用な知識は極力カットした、初心者向き入門書。

Xウィンドウ・システム・シリーズ 日本語版 第7巻
XView Ver.3プログラミング・マニュアル
XView Ver.3リファレンス・マニュアル

Dan Heller 著
予価 6,000円
予価 4,000円

米国O'Reilly & Associates,Inc.から出版されているシリーズの翻訳書。第7巻は、UNIX上で稼働するX11リリース4に対応のXViewツールキット、バージョン3の解説書です。



【好評発売中!】
X11 Release 5 増補版
David Flanagan 著/Adrian Nye 編　大木敦雄監訳　定価4,000円

第4巻　Xツールキット・イントリンシクス・プログラミング・マニュアル
第0巻　Xプロトコル・リファレンス・マニュアル
定価各5,000円

定価は税込

満開の電子ちゃん

作・え 岡村祭

講読方法:定期購読もしくはソフトベンダーTAKERUでお買い求めいただけます。
★定期購読の場合=購読料6ヶ月分6,000円(送料サービス、消費税込)を、
現金書留または郵便振替で下記の宛先へお送り下さい。
現金書留の場合:〒171 東京都豊島区要町1-19-3 いさみビル4F ㈱満開製作所
郵便振替の場合:東京 5-362847 ㈱満開製作所
● ご注文の際は、郵便番号・住所・氏名・電話番号を忘れずに記入して下さい。
● 3.5インチディスク版をご希望の方は、「3.5インチ版」とご指定下さい。
● 新規購読の方は「新規」と明記して下さい。なお、特に購読開始号のご指定がない場合は既刊の最新号からお送りいたします。
● 製品の性格上返品には応じられませんが、お申し出があれば定期購読を解約し残金をお返しします。
★TAKERUでお求めの場合=1部につき1,200円(消費税込)です。
● 定期購読版と内容が一部異なる場合があります。御了承下さい。
● お問い合わせ先 TEL(03)3554-9282(月~金 午前11時~午後6時)
(なお、定期購読版のバックナンバーについては定期購読の方のみご注文を承ります)

澤田眞一(奈良県)

恒星日誌宇宙暦1992年わがエンタープライズ号は満開製作所の指令を受け広告の旅にたった。カーク「Mrスポック、当誌の特徴を報告したまえ。」
スポック「起動は容易かつ操作は最良の手段がとられています。」
K「そうか、それなら普及する可能性も大きいな。」
マッコイ「ツールも多く掲載されているはずだが、スポック?」
S「その意見はあなたにしては珍しく論理的ですね。」
K「それで、内容の方についてはどうなんだ。」
S「……非常に非論理的です。」

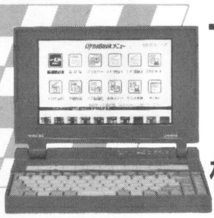
X68000 XVIシリーズ 大特価セール！

AIBIT
アイビット電子株式会社

店舗移転在庫処分セール

	標準価格	特価
CZ-602CBK	￥356,000	￥140,000
CZ-603CGY	￥338,000	￥150,000
CZ-623CTN	￥498,000	￥210,000
CZ-644CTN	￥518,000	特価
CZ-653CBY	￥285,000	￥130,000
CZ-662CBK	￥408,000	￥180,000
CZ-662CGY	￥408,000	￥180,000
CZ-8PK7	￥122,000	￥ 49,800

TOSHIBA
DynaBook EZ
新発売



一太郎dash
&
ロータス1-2-3内蔵
標準価格　￥248,000⇨特価

X68000専用 ハードディスク
HXD-040 特価￥59,000
HXD-042 特価￥64,000
HXD-140(内蔵用) 特価￥65,000

対応機種
CZ-600C CZ-602C CZ-652C CZ-601C CZ-603C CZ-653C

新発売
●書院パソコン
PC-WD1A…………￥330,000
PC-WD1AD…………￥450,000
PC-WD1B…………￥430,000
PC-WD1BD…………￥590,000
●ハイパー電子マネージメント手帳
PV-F1…………￥128,000

CZ-674CH (本体) ￥298,000
CZ-608DH (0.28㎜ディスプレイ) ￥ 94,800
CZ-6FD5 (XVI用外付け 5インチ2ドライブ) ￥ 99,800
標準価格 ￥492,600
35%off 特価￥320,000

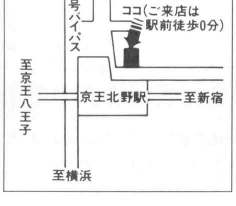
CZ-674CH (本体) ￥298,000
CZ-608DH (0.28㎜ディスプレイ) ￥ 94,800
標準価格 ￥392,800
30%off 特価￥278,000

CZ-674CH (本体) ￥298,000
CZ-606D (0.31㎜ディスプレイ) ￥ 79,800
標準価格 ￥377,800
30%off 特価￥268,000

CZ-674CH (本体) ￥298,000
CZ-606D (0.31㎜ディスプレイ) ￥ 79,800
CZ-6FD5 (XVI用外付け 5インチ2ドライブ) ￥ 99,800
標準価格 ￥477,600
35%off 特価￥310,000

他周辺機器及びポケコン全機種取り扱い。

全商品新品完全保証付〉
・シャープ・シャープ周辺機器（拡張機器全機種、プリンター他）・富士通・NEC常時取り扱い。
・シャープ・カシオポケコン全機種取り扱い。PACIFIC・YHP・キャノンも取り扱い。
・学校、企業納入受け賜ります。送料別料金。★上記商品価格には、消費税は含まれておりません。
・特価表及び資料をご希望の方は、72円切手を同封の上お送りください。
通信販売のお問い合せ、御注文は
TEL.0426-45-3001(本店) FAX.0426-44-6002
営業時間/10:00～19:00●電話受付/9:00～22:00迄可●定休日/水曜日
SHARP SUPER EXE SHOP
アイビット電子株式会社 〒192 東京都八王子市北野町560-5



上記の広告商品はすべて店頭販売もしております。
全国通信販売
北海道から沖縄まで
★送料はご注文の際にお問い合わせ下さい。
★掲載の商品は、すべて新品、保証書付きです。
★掲載の商品は充分用意してありますが、ご注文の際は、在庫の確認の上、現金書留または、銀行振込でお申し込み下さい。全商品クレジットでも扱っております。
★お申し込みの際は必ず電話番号を明記して下さい。
★商品、品切れの節はご容赦下さい。
富士銀行八王子支店　(普)1752505

■店頭にて、新作ゲームソフト25〜30%OFF!!（税別）、超低金利オクトハッピークレジットをご利用下さい!!

■翌月末一括（9月末）払いOK!!手数料無料!!ご利用下さい。■店頭にて、新作ゲームソフト25〜30%OFF!!

パソコンプラザ OCT オクト

オクトで始まるパソコンワールド

☎ 03-3730-6271

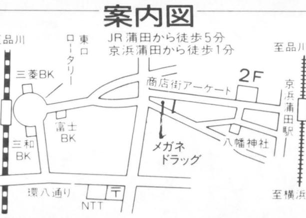
●営業時間 AM 11:00〜9:00/日曜・祭日PM7:00　電話一本で、ハイ即納

〒144 東京都大田区蒲田4-6-7 FAX 03-3730-6273

全国通販

定休日：毎週火曜日
（祭日の場合翌日になります。）

案内図



店頭セール実施中

OCT-1 システム インフォメーション

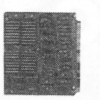
- ▶全商品保証付（メーカー保証）
- ▶超低金利ハッピークレジット（1回〜60回）頭金ナシOK！
- ▶ボーナス一括払いOK！ボーナス 2回・4回・6回 払いOK！
- ▶配達日の指定OK！（万全なサポート体制）
- ▶商品の組合せ自由！ オクトフリーダムシステム
- ▶店頭デモンストレーション実施中

特選周辺機器（送料¥500）

■SX-68MⅡ MIDIインターフェイスボード
（システムサコム）¥19,800… 特価¥13,500

■モデム〔FMMD-311G〕・2400bps.クラス5
（Fujitsu）¥35,800……… 特価¥24,000

■増設RAMボード＝I・Oデータ



①PIO-6BE1-A（1MB）
¥25,000………… 特価¥15,800
②PIO-6BE2-2M（2MB）
¥50,000………… 特価¥31,000
③PIO-6BE4-4M（4MB）
¥88,000………… 特価¥54,000
④SH-6BE1-1M（1MB）
¥25,000………… 特価¥17,800

OCT-1 蒲田 来た〜！待望の夏のセール!! オトクに買って気分爽快!!

SHARP

X68000 PERSONAL WORKSTATION・XVI Compact

■16MHz■
■SX-WINDOW ver1.1■
■Attachment MEMORY BORD■

■CZ-674C-TN（定価¥298,000）

Ⓐ●CZ-674C-TN
●CZ-608D-TN（14型カラーディスプレイ）

定価合計¥392,800▶超特価¥283,000

12回	¥25,900	24回	¥13,700	36回	¥ 9,500	48回	¥ 7,500

Ⓑ●CZ-674C-TN
●CZ-607D-TN（14型カラーディスプレイTV）

定価合計¥397,800▶超特価¥285,000

12回	¥26,100	24回	¥13,800	36回	¥ 9,600	48回	¥ 7,500

Ⓒ●CZ-674C-TN
●CZ-614D-TN（15型カラーディスプレイTV）

定価合計¥433,000▶超特価¥310,000

12回	¥28,300	24回	¥15,000	36回	¥10,400	48回	¥ 8,200

Ⓓ●CZ-674C-TN
●CZ-606D-TN（14型カラーディスプレイ）

定価合計¥377,800▶超特価¥270,000

12回	¥24,700	24回	¥13,100	36回	¥ 9,100	48回	¥ 7,100

※送料¥2,000・税別
（クレジット価格は、送料・税込）

Compact限定セット

【送料¥3,000・税別 クレジットは送料・税込】

①CZ-674CH〈本体〉＋CZ-608DH〈モニター〉
＋CZ-6FD5（5インチドライブ）

定価合計¥492,600▶超特価¥320,000

12回	¥29,300	24回	¥15,500	36回	¥10,800	48回	¥ 8,500

②CZ-674CH〈本体〉＋CZ-606D〈モニター〉
＋CZ-6FD5（5インチドライブ）

定価合計¥477,600▶超特価¥310,000

12回	¥28,400	24回	¥15,100	36回	¥10,400	48回	¥ 8,200

X68000 Compact 大好評記念プレゼント!!

—あなたのオクトから素敵な贈物—

今、Compactをお買い上げいただいた方は、プレゼントの①番か②番のどちらかをお選び下さい。プラス③番は、もれなくプレゼント!!

① F15ストライクイーグルII　大人気　銀河英雄伝説II デラックスセット
新作！大人気ゲームソフト

シュミレーション

（定価¥10,800）

シュミレーション

（定価¥12,600）

or

② インテリジェントコントローラ
■CZ-8NJ2（CYBER STICK）
シューティングゲーマーの必須アイテム!!

（定価¥23,800）

③ ●MF-2HD（5枚）
●シリコンキーボードカバー
もれなく!! サービス!!

※どちらかお選び下さい!!
（どっちが得かヨーク考えてネ！）

ハードディスク（送料¥1,000）

■システムサコム SCSI
●HD-J040
¥ 89,000 42M/25ms……大特価¥ 61,000
●HD-J100
¥128,000 100M/20ms……大特価¥ 87,000
●HD-J130
¥148,000 130M/20ms……大特価¥101,000
●HD-J170
¥189,800 173M/20ms……大特価¥123,000

■ロジテック SCSI
●LHD-FM100E
¥ 99,800 100M…………大特価¥ 69,000
●LHD-FM200E
¥138,000 200M/17ms……大特価¥ 95,000

■エニックス SCSI
●EFX-100B
¥118,000 100M/19ms…大特価TEL下さい
●EFX-140B
¥138,000 140M/16ms…大特価TEL下さい

周辺機器コーナー（送料¥500）

●CZ-6BE2A 2MB RAM（CZ-634C/644C用）……（¥ 59,800）▶特価¥42,500
●CZ-6BE2B 2MB RAM（CZ-634C/644C/674C用）……（¥ 54,800）▶特価¥39,200
●CZ-6BE2D 2MB RAM（CZ-674C用）……（¥ 54,800）▶特価¥39,000
●CZ-6BE2 2MB RAM……（¥ 79,800）▶特価¥59,000
●CZ-6BE4C 4MB RAM……（¥ 98,000）▶特価¥73,000
●CZ-6BFI 増設用RS-232ボード……（¥ 49,800）▶特価¥35,800
●CZ-6BGI GP-IBボード……（¥ 59,800）▶特価¥42,800
●CZ-6BMI MIDIボード……（¥ 26,800）▶特価¥19,200
●CZ-6BNI スキャナ用パラレルボード……（¥ 29,800）▶特価¥21,500
●CZ-6BPI 数値演算プロセッサボード……（¥ 79,800）▶特価¥57,000
●CZ-6BOI ユニバーサルI/Oボード……（¥ 39,800）▶特価¥29,800
●CZ-6EBI/BK 拡張I/Oボックス……（¥ 88,000）▶特価¥66,000
●CZ-6VTI/BK カラーイメージ・ユニット……（¥ 69,800）▶特価¥52,000
●CZ-8NM2A マウス……（¥ 6,800）▶特価¥ 5,100
●CZ-8NTI マウストラックボール……（¥ 9,800）▶特価¥ 7,300
●CZ-8NSI カラーイメージスキャナ……（¥188,000）▶特価¥132,000
●CZ-6BCI FAXボード……（¥ 79,800）▶特価¥ 57,000
●CZ-8TM2 モデムユニット……（¥ 49,800）▶特価¥ 37,000
●LC-10CI-H カラー液晶ディスプレイ……（¥ 59,800）▶特価¥ 45,800
●CZ-6TU GY/BK RGBシステムチューナー……（¥ 33,100）▶特価¥ 23,800
●BF-68PRO 高性能CRTフィルター……（¥ 19,800）▶特価¥ 14,500
●CZ-6MOI 光磁気ディスクユニット……（¥450,000）▶特価¥330,000
●CZ-6BSI SCSIインターフェースボード……（¥ 29,800）▶特価¥ 22,000
●CZ-6BL2 LANボード……（¥298,800）▶特価¥219,000
●CZ-6BVI （ビデオボード）……（¥ 21,000）▶特価¥ 15,400
●CZ-6BP2 数値演算プロセッサ……（¥ 45,800）▶特価¥ 34,300
●AN-S100 スピーカーシステム（2本1組）……（¥ 36,600）▶特価¥ 26,300
●JX-220X カラーイメージスキャナー……（¥168,000）▶特価¥120,000
●CZ-6CSI SCSI変換ケーブル……（¥ 12,000）▶特価¥ 8,900

※クレジットの回数は1回〜60回、ボーナス併用などありますのでお電話でお問合せ下さい。

■本体セット：送料無料（注）本体セット以外の周辺機器（プリンター、モデム、HDD等）及びソフトの送料は、北海道・九州地区＝1ケロ¥1500、■その他離島地区は、1ケロ¥2000となります。
※上記料金には、消費税は含まれておりません。消費税が付加されますので、詳しくは、電話でお問合せ下さい。

■便利てす。夜9時まで営業しております。お立寄り下さい。お待ちしております!!

■特に人気のある商品によっては、しばらくお待ち願うことがありますのでご了承下さい!!

■ビッグバーゲンセール実施中!!ゲームソフト(ビジネス)新製品続々入荷中!

X68000 ——堂々のラインアップ!!

■PROII／XVI■

ボーナス〈2回・4回・6回〉払いOK!!手数料無料!!

X68000 XVI エクシヴィ

(送料・消費税込)

超特価¥残念!表示不能!!

X68000XVI ドッカーン!プレゼント!!
—あなたのオクトから素敵な贈物—

(今、XVIをお買い上げいただいた方は、プレゼントの①番か②番のどちらかお選び下さい。プラス③番はもれなくプレゼント!!)

① F15ストライクイーグルII (定価¥10,800)

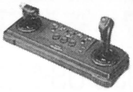
新作!大人気ゲームソフト!!

銀河英雄伝説II デラックスセット (定価¥12,600)

or

※どちらかお選び下さい!!

② インテリジェントコントローラ
■CZ-8NJ2
(CYBER STICK)
シューティングゲーマーの必須アイテム!!
(定価¥23,800)



\+

③ MD-2HD(10枚)／シリコンキーボードカバー　もれなく!!サービス!!

■CZ-634C-TN (定価¥368,000)

Ⓐ●CZ-634C-TN＋CZ-606D-TN
定価合計¥447,800▶**超特価¥表示不能!**

12回	24回	36回	48回
¥27,600	¥14,600	¥10,100	¥ 8,000

Ⓑ●CZ-634C-TN＋CZ-614D-TN
定価合計¥503,000▶**超特価¥表示不能!**

12回	24回	36回	48回
¥31,200	¥16,600	¥11,500	¥ 9,000

■CZ-644C-TN (定価¥518,000)

Ⓒ●CZ-644C-TN＋CZ-606D-TN
定価合計¥597,800▶**超特価¥表示不能!**

12回	24回	36回	48回
¥37,400	¥19,800	¥13,700	¥10,800

Ⓓ●CZ-644C-TN＋CZ-614D-TN
定価合計¥653,000▶**超特価¥表示不能!**

12回	24回	36回	48回
¥40,900	¥21,700	¥15,000	¥11,800

※クレジット表は、送料・消費税込!

X68000PROII ラストチャンス!! 〈BIGプレゼント付〉 (送料無料・税別)

X68000PROII
(CZ-653C)

定価¥285,000
超特価~~¥138,000~~

新作!大人気ゲームソフト!!
銀河英雄伝説II デラックスセット

(定価¥10,800)

プレゼント

さらに!　★JOY CARD(連射式)×2個
さらにさらに!!　★MD-2HD 10枚

■CZ-653C (定価¥285,000)

Ⓐ●CZ-653C＋CU-21HD
定価合計¥433,000▶**超特価~~¥243,000~~**

12回	24回	36回	48回
~~¥21,100~~	~~¥10,500~~	~~¥ 7,000~~	~~¥ 5,200~~

Ⓑ●CZ-653C＋CZ-606D
定価合計¥364,800▶**超特価~~¥198,000~~**

12回	24回	36回	48回
~~¥17,100~~	~~¥ 8,500~~	~~¥ 5,700~~	~~¥ 4,200~~

Ⓒ●CZ-653C＋CZ-607D
定価合計¥384,800▶**超特価~~¥213,000~~**

12回	24回	36回	48回
~~¥18,400~~	~~¥ 9,200~~	~~¥ 6,100~~	~~¥ 4,600~~

Ⓓ●CZ-653C＋CZ-614D
定価合計¥420,000▶**超特価~~¥236,000~~**

12回	24回	36回	48回
~~¥20,400~~	~~¥10,200~~	~~¥ 6,800~~	~~¥ 5,100~~

X68000ソフト大セール実施中!! (ゲームソフト25〜30%OFF) (送料¥500)

〈グラフィック〉●Z's STAFF PRO68K Ver.2.0
(シャフト)定価¥58,000
……………特価**¥36,500**

〈開発ツール〉●CコンパイラPRO-68K Ver.2.1
定価¥44,800 CZ-285LSD
……………特価**¥32,500**

〈統合表計算ソフト〉BUSINESS PRO-68K
Popular 定価¥28,000 CZ-286BSD
……………特価**¥21,000**

〈レイアウト〉●Pressconductor PRO-68K
定価¥28,000 CZ-268BSD
……………特価**¥21,000**

〈C言語〉●C & Professional Pack
定価¥58,000
……………特価**¥39,600**

〈音楽〉●Music studio PRO-68K Ver. 2.0
定価¥28,800 CZ-261MS
……………特価**¥21,200**

〈CGシール〉●CANVAS PRO-68K
定価¥29,800 CZ-249GS
……………特価**¥22,200**

〈ワープロ〉●Multiword Ver.1.1
定価¥32,000 CZ-225BSD
……………特価**¥23,000**

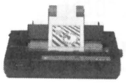
〈OS〉●OS-9 X68000 Ver.2.4
定価¥35,800 CZ-284SSD
……………特価**¥26,900**

型名	商品	定価	特価
CZ-212BS	〈BUSINESS PRO-68K〉	(¥ 68,000)	¥ 48,000
CZ-213MS	〈MUSIC PRO68K〉	(¥ 18,800)	¥ 13,200
CZ-275MWD	〈SOUND SX-68K〉	(¥)	¥TEL下さい
CZ-215MS	〈Sampling PRO-68K〉	(¥ 17,800)	¥ 12,500
CZ-287SS	〈SX-WINDOW Ver.2.0〉	(¥ 12,800)	¥ 9,600
CZ-220BS	〈DATA PRO-68K〉	(¥ 58,000)	¥ 40,000
CZ-272CWD	〈Communication SX-68K〉	(¥ 19,800)	¥ 15,300
CZ-224LS	〈THE 福袋 V2.0〉	(¥ 9,900)	¥ 7,400
CZ-253BS	〈CARD PRO-68K Ver.2.0〉	(¥ 29,800)	¥ 20,800
CZ-258BS	〈Tlepotion PRO-68K〉	(¥ 22,800)	¥ 16,800
CZ-244SS	〈Homan 68K Ver.2.0〉	(¥ 9,800)	¥ 7,500
CZ-247MS	〈MUSIC PRO-68K(MIDI)〉	(¥ 28,800)	¥ 20,800
CZ-240BS	〈Stationery PRO-68K〉	(¥ 14,800)	¥ 11,500
CZ-243BS	〈CYBER NOTE PRO-68K〉	(¥ 19,800)	¥ 15,200

型名	商品	定価	特価
	〈Z's TRIPHY(デジタルクラフト)〉	(¥ 39,800)	¥ 27,000
	〈テラッツオ(ハミングバード)〉	(¥ 19,400)	¥ 13,600
	〈KAMIKAZE(サムシンググッド)〉	(¥ 68,000)	¥ 43,800
	〈Final Ver.3.2(エーエスピー)〉	(¥ 38,000)	¥ 29,000
	〈サイクロンEXPRESSα68〉	(¥ 98,000)	¥ 69,000
	〈Gツール(ザインソフト)〉	(¥ 28,000)	¥ 18,600
	〈たーみのる2(SPS)〉	(¥ 17,800)	¥ 13,000
	〈G68K Ver.2 PRO〉	(¥ 22,000)	¥ 17,300
	〈C-TRACE68 Ver.3.0〉	(¥ 98,000)	¥ 68,500
CZ-251BS	〈ハイパーワード〉	(¥ 39,800)	¥ 29,400
CZ-260LS	〈XBAS to CHECKER PRO-68K〉	(¥ 9.800)	¥ 7,500
CZ-234LS	〈AI-68K〉	(¥188,000)	¥139,000
CZ-255GS	〈CANVASドローグラフィックLIB〉	(¥ 8,800)	¥ 6,600
CZ-256GS	〈CANVASドローグラフィックVol.2〉	(¥ 8,800)	¥ 6,600

プリンタ (送料¥1,000)

■CZ-8PC5-BK
熱転写カラー漢字
定価¥96,800
大特価**¥68,800**



■IO-735X-B
カラーイメージジェット
定価¥248,000
大特価**¥154,000**

今月の推奨品 (送料¥1,000)

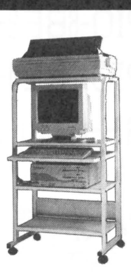
■内蔵用ハードディスク
<Compact XVI(CZ-674C)用>
[KGU-HD80K]
Compact HD-80キット
定価¥168,000

限定特別価格¥TEL下さい!

■5インチフロッピーディスクユニット
<X68000Compact(CZ-674C-H)用>
[CZ-6FD5]
定価¥99,800

限定特別価格¥TEL下さい!

パソコンラック〈送料無料〉



Ⓐ5段キャスター付
スライド式キーボード台
●1150(H)×640(W)×600(D)
定価¥38,000
特価
¥12,500

Ⓑ4段キャスター付
●1250(H)×640(W)×700(D)
定価¥29,800
特価
¥8,800

■ボーナス〈2回・4回・6回〉払いOK!!手数料無料!!ご利用下さい。■店頭にて、新作ゲームソフト25〜30%OFF!!

店頭新作ゲームソフト25〜30%OFF!!ビジネスソフト25%より特価中

★通信販売お申込みのご案内★ 〒144 東京都大田区蒲田4-6-7 TEL:03-3730-6271

お申込みはお電話でお願いします。お客様の〈住所〉〈氏名〉〈電話番号〉及び〈商品名〉をお知らせ下さい。●入金確認後ただちに商品をご送付いたします。

現金一括払い
銀行振込:お近くの銀行より(電信扱い)にてお振込み下さい。
現金書留:封筒の中に住所・氏名・商品名をご記入の上当社までお送り下さい。

クレジット
専用お申込用紙をお送り致します。ので、必要事項をご記入、ご捺印の上ご返送下さい。手続きは簡単です。

オクト ラクラク クレジット表

3回	3.5	6回	4.5	10回	6.0	12回	6.0
15回	9.0	18回	11.0	20回	12.0	24回	12.5
30回	17.0	36回	17.5	48回	23.0	60回	33.0

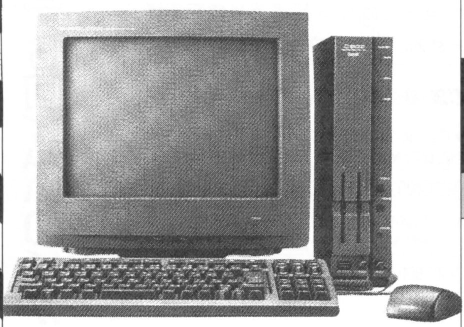
振込先
富士銀行 久ヶ原(クガハラ)支店 ㊔No.1824
三菱銀行 蒲田支店 ㊔No.0278691
株式会社 億人(オクト)

※掲載の価格は変動しますので、まずは、お電話にてご確認ください。
※上記料金には、消費税は含まれておりません。消費税が付加されますので、詳しくは電話でお問合せ下さい。
※銀行振込、または、現金書留でご注文の際には、あらかじめ電話でご確認の上、お申し込み下さい。

マイコンショップ川口

☎0482-25-1718

（消費税別）



New X68000
COMPACT XVI
~~¥298,000~~

CZ-674C-H…………¥298,000
CZ-608D-H…………¥ 94,800
AV-090-SC…………¥168,000

定価 ¥560,000
超 特 価

ソフト各種超特価ご奉仕中

CZ-219SS OS-9/X68000……定価¥29,800
CZ-213MS MUSIC PRO68K 定価¥18,800
CZ-214MS SOUND PRO68K 定価¥15,800
CZ-215MS Sampling PRO68K…………定価¥17,800
CZ-220BS DATA PRO68K…定価¥58,000
CZ-224LS The 福袋 Ver2.0…定価¥ 9,980
CZ-225BS Multiword……定価¥32,000
CZ-251BS Hyper word……定価¥39,800

中古売買価格表

品　名	買取り価格	売　価
CZ-633C	160,000より	180,000より
CZ-644C	210,000より	230,000より
CZ-613C	105,000より	125,000より
CZ-603C	75,000より	95,000より
CZ-612C	85,000より	98,000より
CZ-602C	65,000より	85,000より
CZ-653C	75,000より	95,000より
CZ-663C	95,000より	115,000より
CZ-662C	75,000より	98,000より
CZ-652C	55,000より	75,000より
CZ-611C	70,000より	89,000より
CZ-601C	45,000より	65,000より
CZ-612D	35,000より	45,000より
CZ-602D	30,000より	39,800より
CZ-603D	20,000より	29,800より
CZ-604D	25,000より	34,800より
CZ-605D	45,000より	55,000より

プリンター

CZ-6VT1…………特価¥
CZ-8PG1…………特価¥
CZ-8PG2…………特価¥
CZ-8PK10…………特価¥
CZ-8NS1…………特価¥
CZ-6BC1…………特価¥
CZ-6BG1…………特価¥
CZ-6BP1…………特価¥
CZ-6BP2…………特価¥

ラムボード

CZ-6BE2A……定価¥59,800…特価¥
CZ-6BE2B……定価¥54,800…特価¥
CZ-6BE2D……定価¥　　…特価¥
CZ-6BE1B……定価¥28,000…特価¥
CZ-6BE2……定価¥79,800…特価¥
CZ-6BE4C……定価¥98,000…特価¥
PIO-6BE1-A…定価¥25,000…特価¥
PIO-6BE2-2M 定価¥50,000…特価¥
PIO-6BE4-4M 定価¥88,000…特価¥
SH-6BE1-1M…定価¥25,000…特価¥

ファイル

CZ-6MO1…………定価¥450,000 特価¥
CZ-64H…………定価¥120,000 特価¥
CZ-68H…………定価¥160,000 特価¥

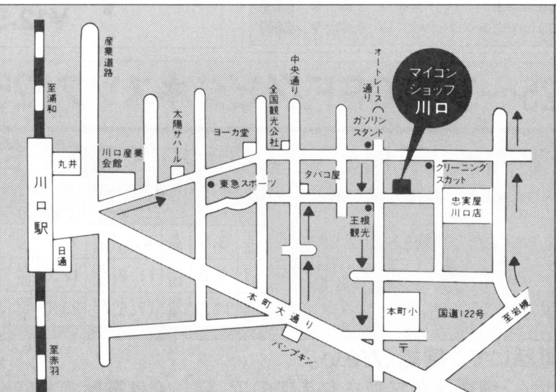
その他機種

CZ-8NS1 カラーイメージスキャナ…定価¥188,000 特価¥
JX-220X カラーイメージスキャナ…定価¥168,000 特価¥
CZ-6BN1 スキャナ用パラレルボード…定価¥ 29,800 特価¥
CZ-6VT1 カラーイメージユニット…定価¥ 69,800 特価¥
CZ-6BV1 ビデオボード……定価¥ 21,000 特価¥
CZ-8TM2 モデムユニット……定価¥ 49,800 特価¥
CZ-8NJ2 インテリジェントコントローラ…定価¥ 23,800 特価¥
CZ-8NM3 マウス・トラックボール…定価¥ 9,800 特価¥
CZ-8NT1 トラックボール……定価¥ 6,888 特価¥
CZ-8NJ1 ジョイカード……定価¥ 1,700 特価¥
CZ-6BC1 FAXボード……定価¥ 79,800 特価¥
CZ-6BM1A MIDIボード……定価¥ 26,800 特価¥
CZ-6BP1 数値演算プロセッサ……定価¥ 79,800 特価¥
CZ-6BP2 数値演算プロセッサ……定価¥ 45,800 特価¥
CZ-6TU-BK-GY RGBシステムチューナー 定価¥ 33,100 特価¥

★クレジット回数1～60回まで設定自由

回　数	1	3	6	12	15	20	24	36	42	48	54	60
金利(%)	2.5	3.5	4.5	6	9	12	12.5	17.5	22	23	28.5	29.5

中古品も取扱っております。



矢印の道路は一方通行です。

通信販売をご利用の方

―全国通販―

通信販売をご利用の方は、売値の変動がありますので在庫、値段をあらかじめ確認のうえ電話で、商品名及びお客様の住所・氏名・電話番号をお知らせ下さい。

絶賛発売中!

CD-ROM Drive for X68000

マルチメディアへの誘い

X68000 Pro SHOP
BASIC HOUSE
KEISOKUGIKEN Corp.
TEL 0286-22-9811　FAX 0286-25-3970

FirstClassTechnology制作のCD-ROM Device Driverを付属させ、ついにX68000用CD-ROM Driveの登場です。本製品を使用することにより、MS-DOSやPC-9801シリーズ、FM-TOWNSなどで採用されている、ISO9660規格のCDをHuman68K/SX-WINDOWで直接扱えるようになります。

また、将来の拡張にも柔軟に対応できるSCSIインターフェースによる接続を採用。ディジーチェーンによって既存のSCSIハードディスクとの同時使用も可能です。

担当　登坂高明

ドライブ仕様

型番	KGU-XCD
使用ドライブ	東芝　XM-3301
平均アクセスタイム	325mSEC
インターフェース	SCSI
キャッシュメモリー	64KB
オーデオ出力	RCA-Phono端子×2 ステレオヘッドホン端子
電源	専用ACアダプター
外形寸法	150×228×50（電源部含まず）

※SCSIケーブル・ターミネーターは別売になります。

付属サポートソフト

ISO9660準拠デバイスドライバ
MusicPlayer for SX-Window
Macintosh™用ファイルビューア for SX-Window

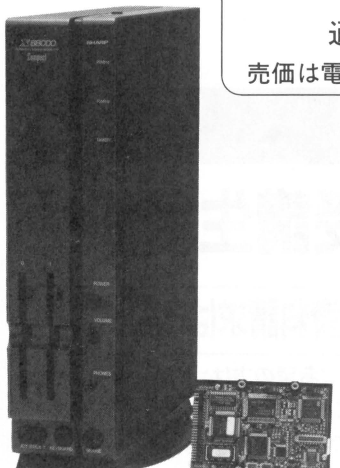
KGU-XCD対応
X68000 CD-ROM第一弾！「フリーウェア集」
Free Soft Ware Selection - CD68K
近日発売
CD-ROM広辞苑検索ユーティリティ for SX-WINDOW
近日発売

標準価格¥118,000-

※Macintosh™はAppleComputerの登録商標です

2.5inch 80MB HDを内蔵 X68000CompactHD

標準価格　¥466,000
通販特価販売中
売価は電話にてお問い合わせください！



ドライブ
Quantum Go-80s™
容量
62MByte
アクセスタイム
16mSEC
インターフェース
SCSI
外部SCSI機器との同時使用可

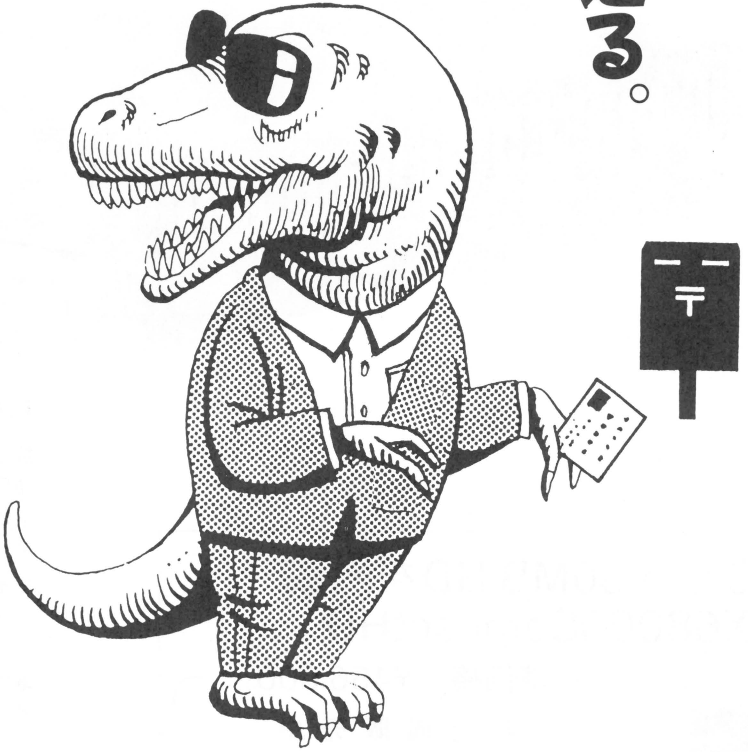
低金利クレジット　通信販売送料 全国一律￥1,000 長期クレジット可能　※表示価格に消費税は含まれておりません

株式会社 計測技研　マイコンショップ BASIC HOUSE 本社／ショールーム／通販部
〒321 栃木県宇都宮市竹林町503-1
TEL 0286-22-9811　FAX 0286-25-3970

NOVE

いよいよ〆切り迫る。
迷っているのは
キミだけかも。
さア！決めた、
進路はゲームデザイナー。

野邊ゲームデザイナーズアカデミー第2期受講生募集は、いよいよ平成4年9月30日で〆切りです。お申し込みはお早めに！



先着500名様！
ガイドフロッピー無料進呈！

ガイドフロッピーはSHARP X68000全シリーズ用とPC9801（VM以降）シリーズに対応のN88日本語BASIC（86）用との2種類を用意してあります。
なお、メディアは5インチです。

自宅でできるゲームデザイナー養成講座

「野邊ゲームデザイナーズアカデミー」受講生募集！

「ゲーム創りを自分の手で」こんな熱意が巷に沸騰中。この期をとらえ、野邊ゲームデザイナーズアカデミーは『コンピュータゲームのノウハウを通信教育で…』を全面に押し出して、ゲームデザイナー養成の道を拓きました。さあ、意欲は持っているのにチャンスに恵まれなかった皆さん、いまこそ全員集合です。

なんでもお問い合せ ☎03(3280)0743

※お問い合わせ受付時間／AM10:00～PM8:00　（土・日・祝は休み）

ガイドフロッピー&資料請求はこちら！

※ガイドフロッピー＆資料請求をご希望の方は、住所、氏名、年齢、職業、電話番号と持っているパソコンの機種名、ご覧の雑誌名を明記の上、ハガキでお申し込み下さい。

〈宛先〉　〒150 東京都渋谷区恵比寿2-32-23
NOVE GAME DESIGNER'S ACADEMY
野邊ゲームデザイナーズアカデミー

SHARP

コンピューター事業拡張につき
プログラマー募集！

提供するのは、X68000の才能をひき出す仕事です。

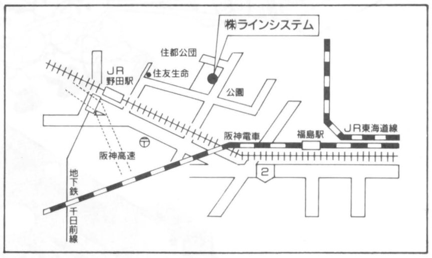
勤務地 大阪・東京・岡山
(男女不問・現地面接可)

■会社概要
設　　立■昭和44年
資 本 金■1,500万円
従業員数■17名
平均年齢■26歳
■事業内容
パーソナルコンピュータ・AXによる自社ソフトパッケージの開発及びオーダーメイド販売サポート
X68000による画像作成業務
資　　格■高卒以上30歳位迄の方
※未経験者歓迎
給　　与■経験・能力等与慮の上、当社規定により優遇いたします。例 25歳　㊊ 176,000円
※別途報奨金制度あり
待　　遇■昇給年1回・賞与年2回 手当/業務・営業・皆勤 交通費全額支給
勤務時間■9:00～18:00
福利厚生■各種社会保険完備 退職金制度 財形貯蓄制度 社内旅行有

経験の有無を問わず、X68000大好き人間 歓迎。経験者には、実力を発揮する場を、未経験者には丁寧な指導をお約束します。
シャープ、XEROX等のシステム機器販売から、シャープ・コンピューターのシステムプレゼンテーターとしてメーカーの期待を担う当社で活躍して下さい。

株式会社ラインシステム
本社　〒553 大阪市福島区鷺洲3丁目1 TEL06-458-7313　担当 菊田
〒115 東京都北区浮間3-2-16 エスポワール403 TEL03-5994-2087　担当 鈴木

休日休暇■隔週休2日制(完全週休2日制も検討中)
祝日
有給・特別・夏期・年末年始休暇等
応　　募■電話連絡の上、履歴書(写真貼付)を持参又は郵送して下さい。追って詳細を連絡いたします。
※入社日相談に応じます。
※応募の秘密厳守いたします。
交　　通■阪神、地下鉄野田駅下車 徒歩7分



GNU ツールボックス
UNIXからDOSへ──X68000の移植を通して
吉野智興 村上敬一郎　共著
B5変型判/240ページ　定価2,200円(税込)

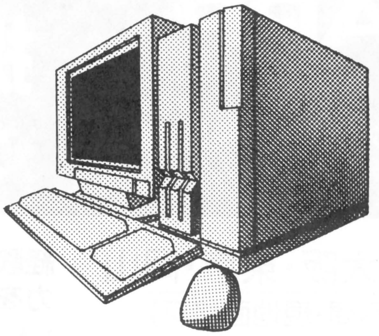
g++、gcc、Nemacsを、X68000に移植するその経緯とノウハウを紹介。68000系のマシンだけでなく、8086系のマシンへの移植も可能です。プログラムをUNIXからDOSへ移植しようとしている方に贈る一冊！

お近くの書店でお求め下さい
ソフトバンク出版事業部 SOFT BANK

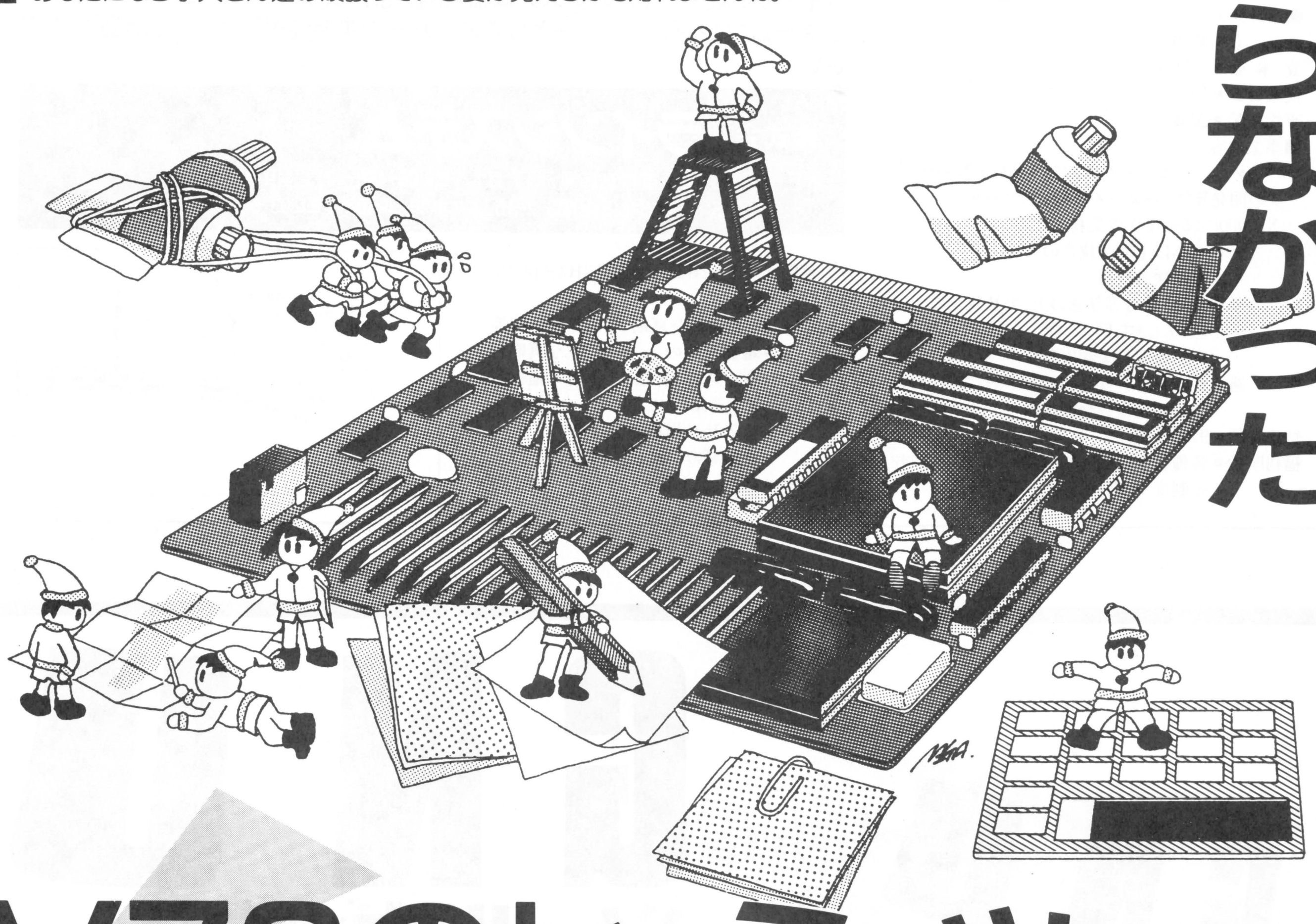
ACCESS

for X68000

V70アクセラレータがどうして速いのか
その秘密をこっそり教えて上げましょう。
それは賢い小人さんがたくさん住んでいるからなのです。
この小人さん達はお絵描きが大好きです。
グラフィック演算はまかせなさい！
レイトレーシングだってあっという間にほらってきあがり。
V70とX68000との面倒なやりとりもみーんなやってくれるので
材料さえ渡しておけばあら簡単！
靴屋のおじいさんのように「眠っている間に」ではなく、
あなたが「お茶を飲んでいる間に」出来上がっているというわけです。
あなたにならい小人さん達の頑張っている姿が見えるかも知れませんね。

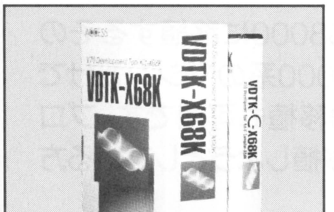


誰も知らなかった



V70のヒ・ミ・ツ

V70 アクセラレータ　VDTK-X68K



※製作：ボード……………有限会社アクセス
ソフトウェア………株式会社ハドソン

VDTK-X68Kの仕様

- ●V70 CPU(μPD70632)
  20MHz 32ビットマイクロプロセッサ
- ●V70AFPP(μPD72691)
  フローティング・ポイント・プロセッサ
- ●メインメモリ(DRAM)2Mバイト
  同一ページ内のアクセスはNo Wait
- ●共有メモリ(SRAM)128Kバイト
  X68000との通信用
- ●併行動作　X68000とV70は、併行に動作することが可能。
  データの受け渡し処理のために双方向ハンドシェークI/Oポートを搭載。

同梱ソフトウェア

- ●アセンブラ
- ●リンカ
- ●ソースコードデバッガ
- ●システムモニタ
- ●フロートエミュレータ
- ●コマンドシェル

オプションソフトウェア

- ●Cコンパイラ
  (VDTK-C-X68K)

価格

●ボードパッケージ　(XVI対応)
VDTK-X68K……………………¥248,000

●オプションソフト (Cコンパイラ)
VDTK-C-X68K……………………¥68,000

購入方法

上記商品は当面の間、通信販売のみとさせて頂きます。
購入ご希望の方は、住所、(社名、所属)氏名、電話番号をお知らせ下さい。注文書をお送りいたします。

●社員募集のお知らせ：アクセスでは技術者を募集しております。担当(田村)までご連絡下さい。

有限会社アクセス
〒101 東京都千代田区神田神保町1-64
神保町協和ビル7F
☎03(3233)0200(代)　FAX.03(3291)7019

資料請求券
oh!X
9月号

パソコン/ワープロ通信ネットワークサービス

J&P HOT LINE

タイムトラベルシリーズ

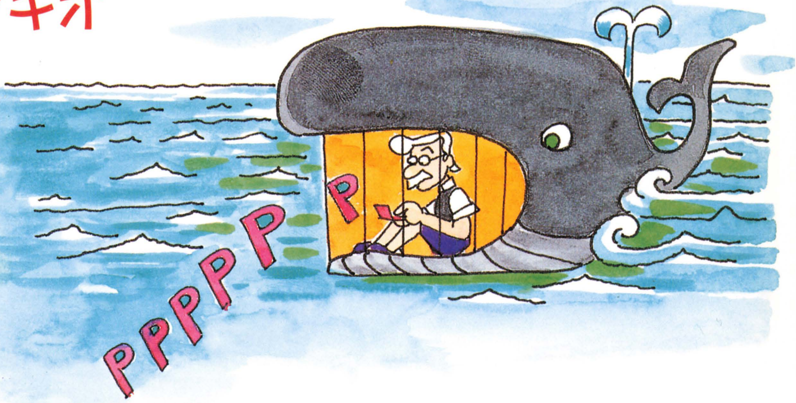
第6回

《もし、この時代にパソコン通信があったなら》

親子の愛は、7つの海をかけめぐる。

ピノキオ

あやつり人形のピノキオは、ある時、そのつくり手ゼペットじいさんの「子供が欲しい」という願いから、妖精の手で「自分で動く」あやつり人形になりました。よろこんだおじいさんの愛情からピノキオは学校へ。でも、悪いキツネたちにそそのかされて、サーカスへと入団してしまいます。まわりの言葉にまどわされながら、人間への道を模索するあやつり人形ピノキオの冒険は、妖精に鼻をニョキニョキ伸ばされたり、あそびの島に迷いこんでロバにされかけたりしながらも、「おじいさんのもとへ帰りたい」というピノキオの熱い気持ちから、広く大きな海の中へと、冒険の舞台を広げるのでした。



もし、この時代にパソコン通信があったなら……

世の中の流行や、うまい話についのってしまう。悪い仲間にそそのかされるピノキオは、まさに私たちの姿そのものかもしれません。もしこの時代にパソコン通信があれば、ピノキオだって悪いキツネの意見だけでなく、もっと広い観点からの意見を参考にできたでしょう。なにより、ゼペットじいさんが、世界の果てまで自分を探しに出かけているということを、ネットワーク上の世間話から知ることができ、もっと早くに本物の人間になれていたことでしょう。

パソコン通信なら、こんな楽しさ。

パソコン通信では、さまざまな職種、さまざまな生き方をしている人が肩書きにとらわれずに話し合っています。思いがけない出会いや、広い視野。あなたも「あやつり人形」ではない、自分の世界を見つけませんか?パソコン通信だから手に入る楽しさですね。

J&P HOT LINEへの
ご入会はスタータキットで。

買ったその日から
2週間無料で
アクセスできます。

お求めは、下記のお店へ。又は現金書留にて、¥3,000+¥90(消費税3%)=¥3,090を事務局までお送り下さい。
すぐにスタータキットをお送りします。

お問い合わせは
〒556 大阪市浪速区日本橋西1-6-5 上新電機株式会社
J&P HOTLINE事務局宛 TEL.(06)632-2521

スタータキットのお求めはJ&P各店でどうぞ。

渋谷店	東京都渋谷区道玄坂2-28-4	☎(03)3496-4141
町田店	東京都町田市森野1-39-16	☎(0427)23-1313
八王子店	東京都八王子市旭町1-1八王子そごう7F	☎(0426)26-4141
立川店	東京都立川市幸町4-39-1	☎(0425)36-4141
三鷹店	東京都三鷹市野崎1-20-17	☎(0422)31-6251
横浜店	横浜市西区北幸2-9-5横浜HSビル1F	☎(045)313-6711
焼津インター店	静岡県焼津市越後島385	☎(054)626-3311
富山店	富山市掛尾町300	☎(0764)22-5033
金沢店	金沢市入江2-63	☎(0762)91-1130
寺地店	金沢市寺地2-3	☎(0762)47-2524
大須店	名古屋市中区大須4-2-48	☎(052)262-1141
テクノランド	大阪市浪速区日本橋5-6-7	☎(06) 634-1211
メディアランド	大阪市浪速区日本橋5-8-26	☎(06) 634-1511
コスモランド	大阪市浪速区難波中2-1-17	☎(06) 634-3111
U. S. LAND	大阪市浪速区日本橋4-9-15	☎(06) 634-1411
ビジネスランド	大阪市北区梅田1-1-3大阪駅前第3ビルB2	☎(06) 348-1881
高槻店	高槻市高槻町11-16	☎(0726)85-1212
くずは店	枚方市楠葉花園町15-2	☎(0720)56-8181
千里中央店	豊中市新千里東町1-3 SENCHU PAL 2番街4F	☎(06) 834-4141
摂津富田店	高槻市大畑町24-10	☎(0726)93-7521
寝屋川店	寝屋川市緑町4-20	☎(0720)34-1166
枚方バイハス店	枚方市田口3-41-7	☎(0720)48-1211
藤井寺店	藤井寺市岡2-1-33	☎(0729)38-2111
岸和田店	岸和田市土生町2451-3	☎(0724)37-1021
さんのみや1ばん館	神戸市中央区八幡通3-2-16	☎(078)231-2111
西宮店	西宮市河原町5-11	☎(0798)71-1171
伊丹店	伊丹市昆陽池1-63	☎(0727)77-5101
姫路店	姫路市東延末1-1住友生命姫路南ビル1F	☎(0792)22-1221
京都寺町店	京都市下京区寺町通仏光寺下ル恵比須之町549	☎(075)341-4411
京都近鉄店	京都市下京区烏丸通七条下ル東塩小路町702	☎(075)341-5769
和歌山店	和歌山市元寺町4-4	☎(0734)28-1441
和歌山南店	和歌山市中島368	☎(0734)25-1414
学園前店	奈良市学園北1-8-10	☎(0742)49-1411
奈良1ばん館	奈良市三条町478-1	☎(0742)27-1111
新大宮店	奈良市法華寺町83-5	☎(0742)35-2611
郡山インター店	大和郡山市横田693-1	☎(07435)9-2221
熊本店	熊本市手取本町4-12	☎(096)359-7800

SHARP

目の付けどころが、
シャープでしょ。

いわば“感性”専用。

ことマインドに関しては
「汎用」という概念は存在しないも同じです。
「実用的である」のと、これなら「使える」というのも違います。
X68000が、普通のパソコンとは違うといわれる所以もここにあります。
いわゆる実用性を重視したビジネスパソコンとは
創造力で一線を画しています。
何に使うのか、何がしたいのか、
パソコン選びのポイントは目的にあったマシンを探すこと。
普通のパソコンに合わせるのでは
あなたのせっかくの創造力も発揮されません。
X68000は、使う人のクリエイティブマインドを咲かせる
“感性”専用パソコンです。

X68000
PERSONAL WORKSTATION・XVI
Compact

本体＋キーボード＋マウス
2HD3.5インチFDDタイプ CZ-674C-H（グレー） 標準価格298,000円（税別）
14型カラーディスプレイ（ドットピッチ0.28mm）
CZ-608D-H（グレー） 標準価格94,800円（税別）

- 5.25インチ増設用フロッピーディスクドライブ CZ-6FD5 標準価格99,800円・税別〔接続ケーブル同梱〕
- ディスプレイテレビ/CZ-6TU用RGBケーブル CZ-6CR1 標準価格4,500円・税別
- ディスプレイテレビ/CZ-6TU用テレビコントロールケーブル CZ-6CT1 標準価格5,500円・税別
- SCSI変換ケーブル CZ-6CS1 標準価格12,000円・税別

（カラー液晶ディスプレイとの組み合わせ例）
10.4型TFTカラー液晶ディスプレイ
LC-10C1-H（グレー）標準価格 598,000円（税別）
接続ケーブル AN-1515X 標準価格 4,200円（税別）

※カラー液晶ディスプレイを接続してご使用の場合、SX-WINDOW上のアプリケーション利用に限定されます。

●お問い合わせは…
シャープ株式会社 電子機器事業本部システム機器営業部〒545大阪市阿倍野区長池町22番22号 ☎(06)621-1221（大代表） 電子機器事業本部AVCシステム事業推進室〒162東京都新宿区市谷八幡町8番地 ☎(03)3260-1161（大代表）

T1002179090607 雑誌 02179-9